

"Express Mail" mailing label number:

EL708268375US

## IMAGE QUALITY TESTER

Clarence E. Rash  
Sheng-Jen Hsieh  
Thomas H. Harding  
Howard H. Beasley  
John S. Martin  
Ronald W. Reynolds  
Robert M. Dillard

### CROSS REFERENCED TO RELATED APPLICATION

The present invention is related to the subject matter of the provisional United States Patent Application entitled, "Field-Ready Monocular Helmet Mounted Display Imagery Evaluation System," naming inventors Clarence E. Rash, Thomas H. Harding, Sheng-Jen Hsieh, Howard H. Beasley, John S. Martin, Ronald W. Reynolds, and Robert M. Dillard, filed October 11, 2000, Attorney Docket No. M-9582 V1. Applicants hereby claim the benefit under 35 U.S.C. §119(e) of the foregoing-referenced provisional application. The contents of the foregoing-referenced provisional patent application are hereby incorporated by reference herein in its entirety.

### BACKGROUND OF THE INVENTION

#### Field of the Invention

The present application relates, in general, to methods and systems related to testing of display sighting systems.

#### Description of the Related Art

A display sighting system, in general, is a system which allows a pilot to integrate himself with a vehicle he is piloting. A display sighting system will generally have one or more unique features depending upon the environment in which the display sighting system is deployed. One feature which a display sighting system might have is the capability to allow a human pilot to see beyond the limitations of normal human vision (e.g., forward looking infrared radar (FLIR) systems, which use radar, image intensification, and infrared waves (which ordinarily cannot be seen by a human pilot) to construct and project a picture which a human pilot can see). Another

feature which a display sighting system might have is the integration of the display sighting system with weapons control of a helicopter (e.g., integrated such that the visual display of the display sighting system is integrated with the gun sights of the weapons on the helicopter).

5 One example of a display sighting system is the Integrated Helmet and Display Sighting system (IHADSS) of the U.S. Army's AH-64 Apache helicopter. (For ease of understanding, the present discussion will refer throughout to the IHADSS, but it will be recognized by those having ordinary skill in the art that the IHADSS is intended to be representative of the more general display sighting systems  
10 referenced above.)

The IHADSS typically gathers information related to the terrain and environment in which the AH-64 is operating by using cameras and/or sensors affixed to the AH-64. Thereafter, the IHADSS processes the gathered information into a form which can be seen by a human pilot, and thereafter projects the gathered and  
15 processed information via an assortment of electronic and optical apparatuses (described in more detail, below) into a human pilot's field of view. In many instances, a pilot of an AH-64 is actually flying the helicopter or targeting the helicopter's weapons systems on the basis of what is displayed by the IHADSS. Accordingly, it is imperative that each individual IHADSS project a clear and  
20 accurate depiction of the terrains and/or environments captured by its associated cameras and sensors. Unfortunately, the integration of each individual IHADSS with the systems and subsystems of AH-64 helicopters in which each individual IHADSS is deployed makes it difficult to ensure that each individual IHADSS is projecting a clear, accurate, and quality depiction of the terrains and/or environments captured by  
25 its associated cameras and sensors. This difficulty is due in large part to a lack of effective methods and systems for the assessment of the accuracy and quality of IHADSS imagery in a field environment.

At present, when an IHADSS is deployed in a field environment, the accuracy and quality of the deployed IHADSS imagery is determined on a subjective basis by  
30 each pilot viewing the IHADSS' display device. Those having ordinary skill in the art will recognize that such a methodology is suboptimal for several reasons. One

[illegible]

15

25

is generally not appropriate for field testing. Yet another reason why testing done in the laboratory environment is generally not adaptable to operation in the field is that it is generally not practicable to deploy the large and bulky testing equipment used in the lab into the field environment.

5 In light of the fact that there is at present no practicable way to adapt IHADSS testing done in the laboratory environment to field environments, it is therefore apparent that the need exists for a method and system which will practicably allow the objective assessment of the functioning of individual IHADSSs in a field environment.

## 10 SUMMARY OF THE INVENTION

The inventors have devised a method and system which will practicably allow the objective assessment of the functioning of individual display sighting systems, such as IHADSSs, in a field environment.

15 In one implementation, a method includes but is not limited to capturing an image actually displayed via a display sighting system; computing at least one difference between the captured image and a recalled representation of the image theoretically displayed via the display sighting system; and presenting the computed at least one difference via a visual display device. In various implementations, circuitry is used to effect the foregoing-described method; the circuitry can be  
20 virtually any combination of hardware, software, and/or firmware configured to effect the foregoing-described method depending upon the design choices of the system designer.

The foregoing is a summary and thus contains, by necessity, simplifications, generalizations and omissions of detail; consequently, those skilled in the art will  
25 appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects, inventive features, and advantages of this patent application will become apparent in the non-limiting detailed description set forth below.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

5 Figure 1 illustrates a perspective view of pilot 100 wearing IHU 102

Figure 2 depicts a perspective view of HDU 104 in isolation.

10 Figures 3A, 3B, 3C, 3D and 3E show three different perspective views of positioning device 300, which is an apparatus which allows HDU 104 in holding fixture 390 to be moved between a first position (see Figure 3D) and a second position (see Figure 3E) such that an image projected by HDU 104 onto combiner 106 can be respectively captured by narrow-angle camera 306 and wide-angle camera 308 (see Figures 3D and 3E).

Figure 4 illustrates test pattern 400 which comes built-in with the IHADSS produced by Honeywell, Inc.

15 Figure 5 shows a high-level logic flowchart.

Figure 6A depicts alternate implementations of the high-level logic flowchart depicted in Figure 5.

20 Figures 6B and 6C illustrate pictographic representations of the methodology used by one implementation to compute the centering difference of method step 602 and the angular difference of method step 600.

Figure 7 illustrates alternate implementations of the high-level logic flowchart depicted in Figure 6A.

Figure 8 shows alternate implementations of the high-level logic flowchart depicted in Figure 5.

25 Figure 9 depicts alternate implementations of the high-level logic flowchart depicted in Figure 8.

Figure A1 shows the IHADSS.

Figure A3 illustrates display size.

Figure A5 depicts a configuration in which two cameras face the HMD from different directions.

Figure A7 shows locations of sensors in proposed HMD fixture design.

Figure A9 illustrates how the HMD hardware fixtures, sensors, I/O cable, and DAQCard-DIO-24 card are integrated.

Figure A11 depicts a display screen, continue button, switches open.

Figure A13 shows display screen, image capture module activated.

Figure A15 illustrates screenshots of parameter setting display screens.

Figure A16 shows an original image/test pattern.

Figure A17 depicts the image of Figure A16 after binary processing.

5

HMD.

HMD.

HMD.

HMD.

HMD.

HMD.

HMD.

10

HMD.

HMD.

HMD.

HMD.

15

HMD.

HMD.

HMD.

HMD.

20

HMD.

Figure B21 illustrates overall testing results of an HMD.

Figure B22 shows tilted test pattern before and after Sober edge detection.

Figure B23 depicts investigation of CCD image capture arrangement.

Figure B24 illustrates a preliminary computer aided design (CAD) concept of  
5 a hardware prototype design.

The use of the same reference symbols in different drawings indicates similar or identical items.

### **DESCRIPTION OF THE PREFERRED EMBODIMENT(S)**

As described above, the U.S. Army's AH-64 Apache helicopter incorporates a  
10 monocular helmet mounted display (HMD) known as the Integrated Helmet and Display Sighting System (IHADSS). The IHADSS consists of various electronic components and a helmet/display system called the Integrated Helmet Unit (IHU).

With reference to the figures, and in particular with reference now to Figure 1,  
shown is a perspective view of pilot 100 wearing IHU 102. Depicted is that IHU 102  
15 includes Helmet Display Unit (HDU) 104. HDU 104 serves as an optical relay device which conveys an image formed on a mini-CRT through a series of lenses (the mini-CRT and lenses are internal to HDU 104, and hence are not shown explicitly), off  
beamsplitter (combiner) 106, an into pilot 100's right eye. Combiner 106 is so named  
in that its construction allows pilot 100 to see whatever image is projected from HDU  
20 104 superimposed, or combined with, whatever pilot 100 can see through combiner 106; in effect, combiner 106 functions as a mirror with respect to the projection of the mini-CRT within HDU 104, and a lens with respect to allowing pilot 100 to see what is in front of him. That is, if one looks closely at Figure 1, one can see that pilot 100's  
eye is visible through combiner 106, which means that pilot can see through combiner  
25 106. Consequently, what pilot 100 sees will be a combination of the projection of HDU 104 and what appears in front of pilot 100 (i.e., whatever pilot 100 can see via the lens function of combiner 106).



Referring now to Figure 2, shown is a perspective view of HDU 104 in isolation.

Referring now to Figures 3A, 3B, and 3C, shown are three different perspective views of positioning device 300, which is an apparatus which allows HDU 104 in holding fixture 390 to be moved between a first position (see Figure 3D) and a second position (see Figure 3E) such that an image projected by HDU 104 onto combiner 106 can be respectively captured by narrow-angle camera 306 and wide-angle camera 308 (see Figures 3D and 3E). Further shown in Figures 3B and 3C is spring 350 which has first end 358 in stationary attachment with camera table 360 and second end 362 in mobile attachment t-shaped-end of lever 352. Lever 352 is shown attached to pivot pole 356. A second end (not shown) of lever 352 is pivotably attached to the underside of holding fixture 390 such that when holding fixture 390 is moved back and forth along guide rails 364 and 366, the second end of lever pivotably attached to the underside of holding fixture 390 will pivot such that t-shaped-end of lever 352 will pivot about pivot pole 356.

In operation, when HDU 104 in holding fixture 390 is moved between a first position (see Figure 3D) and a second position (see Figure 3E) by sliding along guide rails 364 and 366, lever 352 pivots about pivot pole 356 and second end 362 of spring 350 slides along t-shaped-end of lever 352 and is such that when holding fixture 390 is in either the first or second position, spring 350 applies pressure through t-shaped-end of lever 352 such that pressure is applied through second end of lever 352 to holding fixture 390 such that holding fixture 390, and hence HDU 104, is held in either that first or second position.

With reference now to Figures 3D and 3E, shown are two different perspective views of HDU 104 in positioning device 300, which is an apparatus which allows HDU 104 to be moved between a first position and a second position such that an image projected by HDU 104 onto combiner 106 can be respectively captured by narrow-angle camera 306 and wide-angle camera 308.

Referring now to Figure 4, shown is test pattern 400 which comes built-in with the IHADSS produced by Honeywell, Inc. The original main purpose of test pattern 400 was to provide Honeywell field engineers with something consistent to view on

the IHADSS when the IHADSS was first deployed to the field. The original secondary purpose of test pattern 400 was to provide pilots with a fixed pattern which the pilots could use to subjectively optimize their display. Consequently, test pattern 400 is substantially always available on virtually every IHADSS systems. (It is to be understood that although the test pattern of the IHADSS is discussed herein for sake of simplicity, the discussion is meant to be representative of test patterns utilized with display sighting systems, and such test patterns can come preloaded from the factory or may be loaded after the fact, and such test patterns may be multiple in number.)

Depicted is that in one embodiment, test pattern 400 is treated as having two portions: wide-angle portion 402, which is co-extensive with test pattern 400 itself, and narrow-angle portion 404, which is denoted as the small rectangular portion substantially between the gray-scale strips in Figure 4. Wide-angle portion 402 and narrow-angle portion 404 are so named because in one embodiment wide-angle camera 308 and narrow-angle camera 306 are used to respectively capture such portions.

With reference now to Figure 5, shown is a high-level logic flowchart. Method step 500 depicts the start of the process. Method step 502 illustrates capturing an image of IHADSS test pattern 400. Method step 504 shows computing at least one difference between the captured image of IHADSS test pattern 400 and a recalled representation of the image of IHADSS test pattern 400 theoretically displayed via the display sighting system; that is, there is stored in memory (e.g., computer memory) a representation of how the image of IHADSS test pattern 400 would or should appear if the IHADSS is both functioning correctly and set up properly (e.g., by a pilot), and it is such a representation that is recalled. Method step 506 shows presenting the computed at least one difference via a visual display device (e.g., via a portable computer system display). Method step 508 shows the end of the process.

Referring now to Figure 6A, shown are alternate implementations of the high-level logic flowchart depicted in Figure 5. Method step 600 illustrates that in one implementation method step 504 includes computing at least one angular difference between an angular orientation of the captured image and the recalled representation

of the image theoretically displayed via the display sighting system. Method step 602 shows that in one implementation method step 504 includes computing at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system.

5 Method step 604 depicts that in one implementation method step 504 includes computing at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system. The remaining method steps of the flowchart depicted in Figure 6 function substantially as shown and described herein.

10 In one instance, the implementations of Figure 6A are carried out by capturing narrow-angle portion 404 of IHADSS test pattern 400 with narrow-angle camera 306, converting the captured narrow-angle portion to "binary form," and recalling a representation of what narrow-angle portion 404 of IHADSS test pattern 400 should look like if the IHADSS system were functioning substantially optimally. As used  
15 herein converting to "binary form" means creating a two-color image, where all captured image pixels having a number below a calculated threshold are set to gray level 0 (pure black) and all captured image pixels above a calculated threshold value are set to gray level 255 (pure white in a system with 0-255 gray levels). Specific examples of the foregoing described conversion to binary images appear in Specific  
20 Implementation A, Design of Interface and Algorithms for an Image Quality Tester (see, especially Figures A16 and A17 and the discussion of same), and Specific Implementation B, Preliminary Design of an Image Quality Tester for Helmet-Mounted Displays.

25 Referring now to Figures 6B and 6C, shown are pictographic representations of the methodology used by one implementation to compute the centering difference of method step 602 and the angular difference of method step 600. Figure 6B shows that, in one implementation of method step 602, the center point of the actual image captured (point "B") can be compared with where the center point should be if the IHADSS were functioning substantially optimally (point "A," which is obtained from  
30 the recalled representation of the image theoretically displayed via the display sighting system), and the resulting x-y displacement used to compute the distance difference,  $d$ , between actual and theoretical center point locations (the x-y units can

be any unit of length, but in one embodiment the x-y units are millimeters). With reference to Figure 6C, shown is that a right triangle drawing on the x-y position relative to points A and B can be used in combination with a point chosen to be on the “vertical” line of captured narrow-angle portion 404 of IHADSS test pattern 400 in order to calculate  $\theta$  (theta) as the orientation difference between the captured image orientation and the theoretically ideal images. Specific examples of the foregoing appear in Specific Implementation A, Design of Interface and Algorithms for an Image Quality Tester (especially in the Algorithm Design section, subsection C, wherein various approaches for identifying the center point and identifying test pattern orientation and displacement are described), and in Specific Implementation B, Preliminary Design of an Image Quality Tester for Helmet-Mounted Displays.

With respect to computing the focus difference of method step 604, note that narrow-angle portion 404 of IHADSS test pattern 400, subsequent to being converted to binary form, has four bright white vertical lines bounded by dark black regions. It has been discovered by the inventors that measurement of luminance differences between the alternating bright white and dark black regions can be utilized to determine focus. For example, in Specific Implementation B, Preliminary Design of an Image Quality Tester for Helmet-Mounted Displays, Figure B12 and its supporting text shows how luminance measurements appear when narrow-angle portion 404 of IHADSS test pattern 400 is in focus. In contrast, Figure B13 and its supporting text shows several examples of how luminance measurements appear when narrow-angle portion 404 of IHADSS test pattern 400 is out of focus. Consequently, in one implementation, the focus difference of method step 604 can be calculated based on how far the measured luminances of the alternating black and white lines of the binary image form of captured narrow-angle portion 404 of IHADSS test pattern 400 varies from the ideal (e.g., such as that appearing in Figure B12).

With reference now to Figure 7, depicted are alternate implementations of the high-level logic flowchart depicted in Figure 6A. Method step 700 illustrates that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer) the at least one angular difference between an angular orientation of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of

method step 600). Method step 702 shows that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one centering difference between a center point of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 602). Method step 704 depicts that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one focus difference between an optical power of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 604). The remaining method steps of the flowchart depicted in Figure 7 function substantially as shown and described herein.

Referring now to Figure 8, shown are alternate implementations of the high-level logic flowchart depicted in Figure 5. Method step 800 illustrates that in one implementation method step 504 includes computing at least one gray-shades- displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system. Method step 802 shows that in one implementation method step 504 includes computing at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system. Method step 804 shows that in one implementation method step 504 includes computing at least one image quality figure of merit indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system. The remaining method steps of the flowchart depicted in Figure 8 function substantially as shown and described herein.

In one instance, one implementation of method step 800 of Figure 8 is carried out by capturing wide-angle portion 402 of IHADSS test pattern 400 with wide-angle camera 308, and thereafter recalling a representation of what wide-angle portion 402 of IHADSS test pattern 400 should look like if the IHADSS system were functioning substantially optimally. The underlying methodology of method step 800 is based on an agreed upon luminance difference which is expressed as square root of two

differences in gray or luminance levels (this standard is a de facto standard which has arisen over the years by course of use). Thus, in one embodiment, the captured image is iteratively moved through and a count is kept as to how many square root of two gray scale “jumps,” or discontinuities (which occur in test pattern 400 at the

5 boundaries of the varying gray bars – see Figure 4), are detected. This number of discontinuities are then compared against what is expected if the IHADSS is operating correctly. Specific examples of the foregoing appear in Specific Implementation A: Design of Interface and Algorithms for an Image Quality Tester (e.g., Figure A15 and Algorithm Design section, subsection D, which discusses identifying the number of

10 gray shades within a test pattern), and Specific Implementation B: Preliminary Design of An Image Quality Tester for Helmet-Mounted displays.

In one instance, one implementation of method step 802 of Figure 8 is carried out by capturing wide-angle portion 402 of IHADSS test pattern 400 with wide-angle camera 308, and thereafter recalling a representation of what wide-angle portion 402

15 of IHADSS test pattern 400 should look like if the IHADSS system were functioning substantially optimally. The underlying methodology of method step 802 is based on a recognition that when the IHADSS is functioning properly, and the pilot has not improperly adjusted display size, the boundaries of the test pattern should be just on the edge of the field of view of the display. Accordingly, if the boundaries are found

20 at the wrong location, or if the boundaries are not detected at all, it is known that there is a malfunction of some sort. Specific examples of the foregoing appear in Specific Implementation A: Design of Interface and Algorithms for an Image Quality Tester (see e.g., Figure A15 and Algorithm Design section, subsection E, which discusses identifying boundary lines), and Specific Implementation B, Preliminary Design of

25 An Image Quality Tester for Helmet-Mounted displays.

In one instance, one implementation of method step 804 of Figure 8 is carried out by capturing wide-angle portion 402 of IHADSS test pattern 400 with wide-angle camera 308, and thereafter recalling a representation of what wide-angle portion 402 of IHADSS test pattern 400 should look like if the IHADSS system were functioning

30 substantially optimally. The underlying methodology is based on a recognition that when the IHADSS is functioning properly, and the pilot has brightness and contrast levels of the IHADSS set within acceptable tolerances, the average gray level detected

should be within certain defined tolerances relative to average gray level drawn on how the image should appear if the IHADSS is functioning in a substantially optimal mode. Accordingly, if the detected gray levels are substantially outside the bounds of the defined tolerances, it is known either that the system is malfunctioning or that the pilot has either the brightness and/or contrast settings wrong. Specific examples of the foregoing appear in Specific Implementation A: Design of Interface and Algorithms for an Image Quality Tester (e.g., Figure A15 and Algorithm Design section, subsection E, which discusses identifying the contrast, brightness, and gray level relationship), and Specific Implementation B: Preliminary Design of An Image Quality Tester for Helmet-Mounted displays.

With reference now to Figure 9, depicted are alternate implementations of the high-level logic flowchart depicted in Figure 8. Method step 900 illustrates that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one gray-shades-displayed difference between gray shades of the captured image and gray shades of the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 800). Method step 902 shows that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one field-of-view difference indicated by a difference between a boundary location of the captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 802). Method step 904 shows that in one implementation method step 506 includes presenting (e.g., via a visual display device of a portable computer system) the at least one image quality figure of merit indicated by a difference between brightness, contrast, and gray level of a captured image and the recalled representation of the image theoretically displayed via the display sighting system (e.g., the computed quantity of method step 804). The remaining method steps of the flowchart depicted in Figure 9 function substantially as shown and described herein.

Those skilled in the art will recognize that the state of the art has progressed to the point where there is little distinction left between hardware and software implementations of aspects of systems; the use of hardware or software is generally

5

10

15

20

25

30



Those skilled in the art will recognize that it is common within the art to describe devices and/or processes in the fashion set forth above, and thereafter use standard engineering practices to integrate such described devices and/or processes into data processing systems. That is, the devices and/or processes described above can be integrated into data processing system via a reasonable amount of experimentation. Figures 7 and 8 show an example representation of a data processing system into which the described devices and/or processes may be implemented with a reasonable amount of experimentation.

and/or processes described herein may be implemented. It should be noted that a graphical user interface systems (e.g., Microsoft Windows 98 or Microsoft Windows NT operating systems) and methods can be utilized with the data processing system depicted in Figure 10. Data processing system 1020 is depicted which includes system unit housing 1022, video display device 1024, keyboard 1026, mouse 1028, and microphone 1048. Data processing system 1020 may be implemented utilizing any suitable computer such as a DELL portable computer system, a product of Dell Computer Corporation, located in Round Rock, Texas; Dell is a trademark of Dell Computer Corporation.





## Table of contents

### Page

5	Introduction .....	22
	Functionality and operating process.....	22
	Communication interface design.....	23
10	Data acquisition card configuration .....	24
	POWER SPECIFICATIONS OF THE DAQ CARD .....	25
	Sensors and DAQ card integration.....	26
15	Integration of sensor status information into software design .....	26
	Design of image capture interface module and other features .....	27
20	Algorithm design.....	28
	Conclusion and future directions.....	31
	References .....	33
25	Appendix A. List of manufacturers.....	34
	Appendix B. Program code .....	35

### List of figures

30	A1. The IHADSS .....	22
	A2. The IHADSS HDU.....	22
35	A3. Display size .....	23
	A4. IHADSS built-in test pattern .....	23
	A5. Previous design.....	23
40	A6. Revised design.....	23
	A7. Location of sensors on proposed HMD fixture design.....	24

Table of contents (continued)  
List of figures (continued)

	<u>Page</u>
5	
A8. Typical DAQCard-DIO-24 configuration .....	24
A9. Schematic diagram of proposed design .....	26
10	
A10. Initial display screen, switches open .....	27
A11. Display screen, Continue button pressed, switches open .....	27
A12. Display screen, switches 1 and 2 pressed .....	27
15	
A13. Display screen, image capture module activated .....	27
A14. Screenshot of image capture module .....	27
20	
A15. Screenshots of parameter setting display screens .....	27
A16. Original image .....	29
25	
A17. Image after binary processing .....	29

List of tables

30	1. Pin assignments for the CB terminal block .....	25
	2. Power specifications for input and output signals .....	26

## Introduction

The U.S. Army's AH-64 Apache attack helicopter incorporates a monocular helmet mounted display (HMD) known as the Integrated Helmet and Display Sighting System (IHADSS). The IHADSS consists of various electronic components and a helmet/display system called the Integrated Helmet Unit (IHU). The IHU (Figure A1) includes a helmet, visor housings with visors, miniature cathode ray tube (CRT), and helmet display unit (HDU). The HDU (Figure A2) serves as an optical relay device which conveys the image formed on the CRT through a series of lenses, off a beamsplitter (called a combiner), and into the aviator's right eye. The CRT is 1 inch in diameter and uses a P-43 phosphor. The combiner is a multilayer dichroic filter which is maximized for reflectance at the peak emission of the P-43 phosphor.

See  
Figure A1. The IHADSS.

See  
Figure A2. The IHADSS HDU.

Currently, there is no existing objective system for validation in the field of the quality of the imagery presented via the IHADSS. To maintain system integrity and readiness, and to provide pilots with validated pilotage, navigation, and fire control imagery, there is a need to develop an image quality testing tool for the IHADSS. This tester could be used as a validation tool to verify settings for regular flight missions and for preventive maintenance tasks. A preliminary tester design for the AH-64's IHADSS HMD was proposed and reported in Hsieh et al. (1999).

The objectives of the work reported here were to design and integrate communications interface and software procedure components for the proposed IHADSS HMD image tester. This is a continuation of the previous effort. The function of the communications interface is to sense (i.e., calculate) the positions of the camera and HMD based on the status of limit switches attached to a fixture used to mount the camera and HMD. This information then is converted to an eight-bit binary value using a digital I/O (input/output) data acquisition card. This discrete value is used by a custom developed software program as an indicator of the status of the hardware. Image capture routines then are activated to capture the test pattern generated by the HMD under test. The program applies image processing procedures to the images. In addition, image processing algorithms that can extract image features from HMD imagery and analyze them relative to the design specifications are proposed. These developments will allow automated evaluation of the image quality of an HMD.

## FUNCTIONALITY AND OPERATING PROCESS

The IHADSS HMD has a 30-degree vertical by 40-degree horizontal field of view (FOV). Corner obscuration is permissible and symmetrical, as illustrated in Figure A3. The built-in test pattern (Figure A4) of the IHADSS HMD is used as the inspection specification on which the tester will be based. The test pattern shows strips of gray opposed along a centering line. Each strip contains a minimum of 8 to 10 shades of gray, depending on the contrast ratio. Adjacent shades have a square

5

**See**

10

**See**

15

- 20

## 25

30

35

See

40

- 45

- Sensor #3 senses the HMD in position #2.
- System captures the image.

5 Three sensors are required to accomplish the above sequence of events. One would be used to indicate the presence of the HMD, a second to indicate that the HMD is facing camera #1, and a third to indicate that the HMD is facing camera #2. Figure A7 shows the position of the sensors with respect to the HMD fixture.

10 See  
Figure A7. Locations of sensors in proposed HMD fixture design.

### DATA ACQUISITION CARD CONFIGURATION

15 A data acquisition card (DAQCard-DIO-24) by National Instruments was chosen to interface between the hardware sensors and the software. This card can fit into either of the notebook computer's PCMCIA slots.

20 In addition, an input/output (I/O) cable and terminal block are available to facilitate connecting the DAQ card to external devices such as panel meters, instruments, and solid-state relays. Figure A8 displays this configuration. Since the proposed tester is driven (in this prototype stage) by a notebook computer to minimize the size of the tester, the ability to use the PCMCIA slot as the I/O interface channel between the sensor hardware and system software was an essential feature.

25 See  
Figure A8. Typical DAQCard-DIO-24 configuration.

30 This DAQCard-DIO-24 provides three configurable ports with 24 available digital I/O lines, which allow it to switch external devices such as transistors and solid-state relays, read the status of external device digital logic, and generate interrupts. Table 1 describes the cable pin assignment to the terminal block. Even numbers are signal grounds and odd numbers are the I/O signal lines. There are eight signals lines associated with each port (e.g., ports A, B, and C). Thus, the eight signal lines associated with port A are denoted as PA0, PA1, and so on, up to PA7. The  
35 same notation applies for ports B and C.



Table 1.

GND	2	1	PC7
GND	4	3	PC6
GND	6	5	PC6
GND	8	7	PC4
GND	10	9	PC3
GND	12	11	PC2
GND	14	13	PC1
GND	16	15	PC0
GND	18	17	PB7
GND	20	19	PB6
GND	22	21	PB5
GND	24	23	PB4
GND	26	25	PB3
GND	28	27	PB2
GND	30	29	PB1
GND	32	31	PB0
GND	34	33	PA7
GND	36	35	PA6
GND	38	37	PA5
GND	40	39	PA4
GND	42	41	PA3
GND	44	43	PA2
GND	46	45	PA1
GND	48	47	PA0
GND	50	49	+5V

## 5

10

**Table 2.**  
Power specifications for input and output signals.

Input signals	Level	Min	Max
	Input logic high voltage	2.2 V	5.3 V
	Input logic low voltage	-0.3 V	0.8 V
	Input current (0 < Vin < +5 V)	-1.0 mA (milliampere)	1.0 mA (milliampere)
Output signals			
	Pin 49 (at +5 V)	--	1.0 A

By default, all digital lines are pulled up to a logical HIGH setting. To keep a digital line in a logical LOW position, a 4.7 k $\Omega$  resistor from the digital line to ground can be connected in parallel with the external device. For example, to pull PC7 down to logical LOW, if the DAQCard-DIO-24 is connected to a CB-50 I/O terminal block (see Figure A7), pin 1 can be connected to any even numbered ground pin on the CB-50 pin I/O connector with a 4.7 k $\Omega$  resistor in between.

#### SENSORS AND DAQ CARD INTEGRATION

As described earlier, three sensors (i.e., limited switches) are used to sense the HMD position and presence. Figure A9 shows how the HMD hardware fixtures, sensors, I/O cable, and DAQCard-DIO-24 card are integrated. A pull down 4.7 k $\Omega$  resistor is utilized for each input signal pin. Input signal pins PA0, PA1, and PA2 are each connected to a limited switch. Reading the return value from the 8-bit I/O signals allows determination of which switch has been pressed. For instance, by default, the return value of an I/O signal is 255, since all the input pins are in logical HIGH position. A return value of 254 indicates that switch 1 has been pressed. If a limited switch is mounted close to camera #1, one can further interpret that HMD is facing camera 1. Figure A9 shows a schematic diagram of the proposed design.

See  
Figure A9. Schematic diagram of proposed design.

#### INTEGRATION OF SENSOR STATUS INFORMATION INTO SOFTWARE DESIGN

The DAQCard-DIO-24 card is used to capture the sensors' status so that the software system can fuse this information with other sequences of events. For instance, knowing the status of limited switch #1 (which is mounted on the bottom of the enclosure) allows the system to determine if the HMD is present or not; and thus whether or not to activate the image capture routines. A Visual Basic function has been designed to query the hardware register that records the sensor status. Figures A10-A13 demonstrate the integration of an HMD setup and image capture modules

using feedback from the designed function. Switch #1, which indicates the presence of an HMD, will be mounted on the bottom of the enclosure. Switch #2, which indicates whether or not the HMD is facing the narrow-angle camera, will be mounted near the stopper on the rack by camera #1. Switch #3, which indicates whether or not the HMD is facing the wide-angle camera, will be installed near the stopper on the opposite side of the rack by camera #2.

See  
Figure A10. Initial display,  
screen, switches open.

See  
Figure A11. Display screen,  
Continue button pressed, switches open.

For details about the specifications and configuration of the DAQCard-DIO-24 card, please refer to the (1) DAQCard-DIO-24 user manual and (2) DAQ user manual for PC compatibles. For details about the code developed for this module, see Appendix B.

See  
Figure A12. Display screen, switches  
1 and 2 pressed.

See  
Figure A13. Display screen,  
image capture module activated.

#### Design of image capture interface module and other features

Revisions to previously developed modules and additional features include the following:

- Image capture interface module: In the previous effort (Hsieh et al., 1999), object-linked embedding (OLE) techniques were used to launch the image capture driver included with the MRT Video-Port Professional software package. Based on Army recommendations, this module has been replaced with a new module written using the software's built-in tool-box library. Only the most essential functions are provided by this revised module. The revised module also provides an image format with a 780 x 510 pixel resolution. Figure A14 is a screenshot of the newly designed image capture interface module, showing an image of the HMD test pattern taken using the wide-angle camera.

See  
Figure A14. Screenshot of image capture module.

See  
Figure A15. Screenshots of parameter setting display screens.

- Password protection for system accessibility: A password is needed to enter the system or change parameter settings.
- Parameter setting features: Some parameters are camera dependent and/or user dependent. A password is needed to change parameter settings. For example, the size of a test pattern is camera lens angle dependent; thus, the gray-shade stripes'

5

## ALGORITHM DESIGN

15

20

- 40

45

A. Identify the number of center lines.

**Step 2. Draw multiple lines across X and/or Y axes.**

**Step 4. Store the intersection points in an array with multiple dimensions.**

**Step 6. Develop regression lines to compare the parallel property.**

where B = black pixel and W = white pixel.

### How to find the threshold value needed to conduct the binary image process:

**Step 2. Calculate the center/horizontal lines in area A.**

Step 4. Find  $\alpha$  knowing that the probability  $p(x \geq \mu + \alpha s) = \gamma/2$ .

Where  $\mu$  is the mean and  $S$  is the standard deviation of the gray level of the image, and  $\gamma$  represents the percentage of the center four-line region relative to the overall image area. The center four lines are the ones that have a higher gray level than the rest of the background;  $\gamma/2$  will provide a better contrast of the center four-line area.

**Step 1.  $P(m, n)$  where  $m = 0, \dots, 780$  and  $n = 0, \dots, 510$**

H: horizontal line, V: four vertical lines, O: center overlap region

A = 33122

Step 4.  $p(x \geq \mu + \alpha s) = 0.0416$ , where  $\mu = 24.14$ ,  $s = 29.67$ ; therefore,  $\alpha = 2.652$

Step 5.  $T = \mu + \alpha s$ ; therefore,  $T = 102.82$

See  
Figure A17. Image after binary  
processing.

689433 v1



**Step 6.** Use the square root of 2 to determine the number of gray shades.

represents the pixel  $p(x, y)$  with the lowest gray level.

689433 v1

Professional image grabber software package. Algorithms were designed, taking into consideration the steps of data collection, design specifications, and noise generation. Three HMD units were utilized to capture image data. Images with noise such as displacement, orientation, and focus were captured. Statistical approaches such as correlation coefficients and regression analysis were utilized to probe the relationship between performance/image variables (such as focus) and image gray level variation. Knowledge of these relationships allows use of image variables to verify and/or predict control variables such as focus resolution.

Image measurement specifications were developed based on statistical analysis of the collected image data. Algorithms for detecting four vertical lines, center point, focus, and boundary are proposed. Examples are given to illustrate how the procedures work and screenshots of the before and after image processing are shown.

Future work will likely include:

- Coding of the designed image specification and algorithms and verification with image data collected from the field.
- Fabrication of the image tester with a robust fixture holder which has three spring loaded jags to provide constant pressure around the HMD and to accommodate variation between HMDs.
- Field evaluation of tester accuracy.







## APPENDIX B.

### Program code.

#### 5 EXIT SETTING FORM

Option Explicit

Private Sub Image2\_Click()

End Sub

Private Sub nolButton\_Click()

password.Show  
Unload exitwnd

End Sub

Private Sub yesButton\_Click()

Unload password  
Unload Me

End Sub

---

#### 820F1 FORM

Private Sub Frame1\_DragDrop(Source As Control, X As Single, Y As Single)

End Sub

Private Sub Timer1\_Timer()

Dim PauseTime, Start

PauseTime = 2 'Set duration.

Start = Timer 'Set start time.

Do While Timer < Start + PauseTime

DoEvents 'Yield to other processes.

Loop

Unload Me

password.Show

End Sub

---

#### DIONEPORTBACKUP FORM

Option Explicit

Option Base 0

'Constant for PrintText

Const LEN\_PRINTTEXT = 4096

```

' *****
' SUBROUTINE: PrintText
' DESCRIPTION: PrintText to desired TextBox (upto 4096 characters)
5 ' INPUTS:   txtBox - TextBox to print on
'           strText - Text to print
' *****
Sub PrintText(txtBox As TextBox, strText As String)

10     txtBox.Text = Right$(txtBox.Text + strText$ + Chr$(13) + Chr$(10), LEN_PRINTTEXT)

    txtBox.SelStart = Len(CStr(txtBox.Text))

    DoEvents
15 End Sub

' *****
' SUBROUTINE: cmdExit_Click
20 ' DESCRIPTION: Clean up and exit
' *****
Sub cmdExit_Click()

    End
25 End Sub

' *****
' SUBROUTINE: Form_Load
30 ' DESCRIPTION: Gets automatically called at startup
' *****
Sub Form_Load()

    If (FlagLabel = 0) Then
35         Step1.BackColor = &H80FF80
        Call PrintText(txtStatusBox, "Place HMD in camera #1 position ! ")
    Else
        Step1.BackColor = &HFFFFFF   White
        Step1.ForeColor = &H800000F   Black
40 End If

    End Sub

' *****
45 ' SUBROUTINE: cmdContinue_Click
' DESCRIPTION: The main NI-DAQ operations are here
' *****
Sub cmdContinue_Click()

50     '
    ' Local Variable Declarations:

    Dim iStatus As Integer
    Dim iRetVal As Integer
55     Dim iDevice As Integer
    Dim iPort As Integer
    Dim iMode As Integer
    Dim iDir As Integer
    Dim iPattern As Long
    Dim iIgnoreWarning As Integer
60     Dim PauseTime, Start

    iDevice% = 1

65     ' Temporarily disable buttons for protection from 'DoEvents'
    cmdContinue.Enabled = False

```

cmdExit.Enabled = False

Configure port as input, no handshaking.

5 iStatus% = DIG\_Prt\_Config(iDevice%, iPort%, iMode%, iDir%)  
iRetVal% = NIDAQErrorHandler(iStatus%, "DIG\_Prt\_Config", iIgnoreWarning%)  
iStatus% = DIG\_In\_Prt(iDevice%, iPort%, iPattern%)  
iRetVal% = NIDAQErrorHandler(iStatus%, "DIG\_In\_Prt", iIgnoreWarning%)

10

\*\*\*\*\*  
'PA0 -- HMD position switch; On/254, Off/255  
'PA1 -- HMD in camera #1 position; On/253, Off/255  
'PA2 -- HMD in camera #2 position; On/251, Off/255  
15 'HMD lucked in camera #1; then, iPattern& = 252  
'HMD lucked in camera #2; then, iPattern& = 250  
'Cover is opened; then, iPattern& = 255  
\*\*\*\*\*

20

If ((iPattern& = 251) Or (iPattern& = 253) Or (iPattern& = 255)) Then  
Call PrintText(txtStatusBox, "Enclosure is open !")  
End If

25

If ((iPattern& = 252) And (FlagLabel = 0)) Then  
Call PrintText(txtStatusBox, "Positon is lucked ! Loading program ")  
Step1.BackColor = &HFFFFFF White  
Step1.ForeColor = &H800000F Black  
30 Step2.BackColor = &H80FF80 Light Green

PauseTime = 2 ' Set duration.  
Start = Timer ' Set start time.  
35 Do While Timer < Start + PauseTime  
DoEvents ' Yield to other processes.  
Loop

40

Unload Me Form1Backup  
Image\_Capture.Show

45

Step1.BackColor = &HFFFFFF White  
Step1.ForeColor = &H800000F Black  
Step2.BackColor = &HFFFFFF White  
Step2.ForeColor = &H800000F Black  
Step3.BackColor = &H80FF80 Light Green

Call PrintText(txtStatusBox, "Place HMD in position indicated for use with wide-angle camera; then press  
Continue button below")

50

FlagLabel = FlagLabel + 1 ' Flag for sequencing the events

End If

55

If ((iPattern& = 250) And (FlagLabel = 1)) Then

Call PrintText(txtStatusBox, "Positon is lucked ! Loading program ")

60

Step1.BackColor = &HFFFFFF White  
Step1.ForeColor = &H800000F Black  
Step3.BackColor = &HFFFFFF White  
Step3.ForeColor = &H800000F Black  
Step4.BackColor = &H80FF80 Light Green

65

PauseTime = 2 ' Set duration.

```

Start = Timer 'Set start time.
Do While Timer < Start + PauseTime
    DoEvents 'Yield to other processes.
Loop

```

5

```

Unload Me
Image_Capture.Show

```

10

```

FlagLabel = FlagLabel + 1

```

```

End If

```

```

If (FlagLabel = 2) Then

```

15

```

    Step1.BackColor = &HFFFFFF 'White

```

```

    Step1.ForeColor = &H8000000F 'Black

```

```

    Step2.BackColor = &HFFFFFF 'White

```

```

    Step2.ForeColor = &H8000000F 'Black

```

```

    Step3.BackColor = &HFFFFFF 'White

```

20

```

    Step3.ForeColor = &H8000000F 'Black

```

```

    Step4.BackColor = &HFFFFFF 'White

```

```

    Step4.ForeColor = &H8000000F 'Black

```

```

    Step5.BackColor = &H80FF80 'Light Green

```

25

```

    Call PrintText(txtStatusBox, "Press Return button below to return to main menu ")

```

```

    FlagLabel = FlagLabel + 1

```

```

End If

```

30

```

If (FlagLabel < 3) Then

```

```

    cmdContinue.Enabled = True

```

```

Else

```

```

    cmdContinue.Enabled = False

```

```

End If

```

35

```

    cmdExit.Enabled = True

```

```

End Sub

```

40

```

Private Sub Image1_Click()

```

```

End Sub

```

45

```

Private Sub Return_Click()

```

```

    If (FlagLabel = 3) Then

```

```

        Step5.BackColor = &HFFFFFF 'White

```

```

        Step5.ForeColor = &H8000000F 'Black

```

```

    End If

```

50

```

    Unload Form1Backup

```

```

    Form2.Show

```

```

End Sub

```

55

## **820F2 FORM**

```

Private Sub Form_Load()

```

60

```

If (DummyY = 0) Then

```

```

    cmdImage_Analysis.Enabled = False

```

```

    cmdResults.Enabled = False

```

```

    cmdSetup.Enabled = True

```

```

Else

```

```

5      End If
      End Sub
      Private Sub cmdSettings_Click()

10      Unload Form2
      passwordforsettings.Show
      cmdSettings.Enabled = False 'once it is set; u can't go back

      End Sub

15      Private Sub cmdSetup_Click()

      cmdSetup.ToolTipText = "Set up the HMD"

20      Unload Form2
      Form1Backup.Show
      DummyY = 1 'to de-activate the functions

      End Sub

25      Private Sub cmdImage_Capture_Click()

      cmdImage_Capture.ToolTipText = "Image capture of the HMD"

      Unload Form2

30      cmdImage_Capture.Enabled = False

      Image_Capture.Show

35      End Sub

      Private Sub cmdImage_Analysis_Click()

      cmdImage_Analysis.ToolTipText = "Image features analysis"

40      Unload Form2
      Form4.Show
      MsgBox "Select an image file"

45      End Sub

      Private Sub cmdResults_Click()

      cmdResults.ToolTipText = "Analysis findings"

50      Unload Form2
      Form5.Show

      End Sub

55      Private Sub cmdQuit_Click()

      cmdQUIT.ToolTipText = "Exit from the system"

60      Unload Form2
      End

      End Sub

65      Private Sub Image1_Click()

```

End Sub

**820F479 00 FORM**

```

5   Public Displacement, Angle As Double
   Public CenterLineSlope As Double
   Public CenterLineIntercept As Double
   Public Center_Point_X, Center_Point_Y As Double

10  Const intUpperBoundX = 780 '320 total
   Const intUpperBoundY = 510 '244 total
   Const n = 4 '# of center line

   Dim X, Y As Integer
15  Dim picObject0, picObject1 As Image 'Do not delete picObject1; U used picObject1 somewhere in the form
   Dim picObject3 As Picture
   Dim Coord_X(0 To 45, 0 To 10) As Integer
   Dim Coord_Y(0 To 45, 0 To 10) As Integer
   Dim pixels(0 To intUpperBoundX, 0 To intUpperBoundY) As Long
20  Dim ImagePixels(2, intUpperBoundX, intUpperBoundY) As Integer

   Private Sub Back_Click()

   cmdBack.ToolTipText = "Back to previous stage"
25  Unload Form4
   Form2.Show

   End Sub

30  Private Sub cmdFocus_Click()

   cmdFocus.ToolTipText = "Focus Measurement"

   *****
35  'Step 1: Calculate the image standard deviation
   'Step 2:
   *****

   Set Picture0.Picture = picObject0
40  For X = 0 To intUpperBoundX - 1
     For Y = 0 To intUpperBoundY - 1
       Picture0.PSet (X, Y), Picture0.Point(X, Y) - 10
     Next Y
   Next X
45

   End Sub

50  Private Sub Form_Load()

   cmdFocus.Enabled = False
   cmdGray_Shade.Enabled = False
   cmdCenter_and_Boundary.Enabled = False
55  cmdEdgeDetection.Enabled = False
   cmdGray_Shade.Enabled = False
   cmdDis_and_Orientation.Enabled = False

   End Sub
60  Private Sub cmdGray_Shade_Click()

   cmdGray_Shade.ToolTipText = "Detecting number of gray shades"

   End Sub

```



```

Private Sub cmdSelect_Click()

Dim filename, EdgeDetection As String
5 Dim bytRed, bytGreen, bytBlue, bytAverage As Integer
Dim GrayLong As Long
Dim SumGrayLevel, MeanGray, SumSquare, StandardDeviation, ThresholdValue As Double

10 cmdSelect.ToolTipText = "Select an image file first"

On Error GoTo FileError
If (Right$(Dir1.Path, 1) = "\") Then
    filename = File1.Path & File1.filename
Else
15 filename = File1.Path & "\" & File1.filename
End If

Open filename For Input As #1

20 Set picObject0 = LoadPicture(filename)
Set Picture0.Picture = picObject0

Close #1

25 'Do not reverse the sequence: image1 and picture0

Open "c:\windows\desktop\ImageMap.txt" For Output As #2

30 For X = 0 To intUpperBoundX - 1
    For Y = 0 To intUpperBoundY - 1

        pixels(X, Y) = Picture0.Point(X, Y)

35 If (pixels(X, Y) = 0) Then
        bytRed = 0
        bytGreen = 0
        bytBlue = 0
    End If

40 If (pixels(X, Y) > 0) Then
        bytRed = GetRed(pixels(X, Y))
        bytGreen = GetGreen(pixels(X, Y))
        bytBlue = GetBlue(pixels(X, Y))
45 End If

        If (Y = 150) Then
            bytAverage = (bytBlue + bytRed + bytGreen) / 3
            Write #2, X, Y, bytRed, bytBlue, bytGreen, bytAverage
50 End If

            'ImagePixels(0, X, Y) = bytRed
            'ImagePixels(1, X, Y) = bytGreen
            'ImagePixels(2, X, Y) = bytBlue
55 'the file u have is in gray scale; therefore, u do not need to average

            Picture0.PSet (X, Y), RGB(bytRed, bytGreen, bytBlue)

        Next Y
60 Next X

Close #2

'SumGrayLevel = 0
65 'For X = 0 To intUpperBoundX - 1
    ' For Y = 0 To intUpperBoundY - 1

```

```

' SumGrayLevel = SumGrayLevel + ImagePixels(0, X, Y)
' Next Y
Next X

5 MeanGray = SumGrayLevel / (intUpperBoundX * intUpperBoundY)
SumSquare = 0

For X = 0 To intUpperBoundX - 1
' For Y = 0 To intUpperBoundY - 1
10 ' SumSquare = SumSquare + ((ImagePixels(0, X, Y) - MeanGray) * (ImagePixels(0, X, Y) - MeanGray))
' Next Y
Next X

StandardDeviation = SumSquare / ((intUpperBoundX * intUpperBoundY) - 1)
15 ThresholdValue = MeanGray + (2.5 * StandardDeviation)

If File1.filename = "Narrow.bmp" Then
cmdEdgeDetection.Enabled = True
20 cmdDis_and_Orientation.Enabled = True
cmdFocus.Enabled = True
MsgBox "Select one of the criterion on the left top corner"
End If

25 If File1.filename = "Wide.bmp" Then
cmdGray_Shade.Enabled = True
cmdCenter_and_Boundary.Enabled = True
MsgBox "Select one of the criterion on the left bottom corner"
End If

30 Exit Sub

FileError: MsgBox "Select an image file first !"

35 End Sub

Private Sub cmdCenter_and_Boundary_Click()

cmdCenter_and_Boundary.ToolTipText = "Finding the center lines and boundary"
40 Set Picture0.Picture = picObject0
For X = 0 To intUpperBoundX - 1
For Y = 0 To intUpperBoundY - 1
Picture0.PSet (X, Y), Picture0.Point(X, Y)
45 Next Y
Next X

Set picObject3 = Picture0.Picture
SavePicture picObject3, "TEST1.BMP"
50 LoadPicture ("TEST1.BMP")

MsgBox "Boundary routine ended !"

End Sub

55 Private Sub cmdEdgeDetection_Click()

Dim RGBLong As Long
Dim G_X, G_Y, G_X_Y As Integer
60 Dim bRXY, bRXm1Y, byRXYm1, bRXm1Ym1 As Integer
Dim bRXp1Y, bRXYp1, bRXp1Yp1, bRXp1Ym1, bRXm1Yp1 As Integer
Dim bytRed, bytGreen, bytBlue As Integer

cmdEdgeDetection.ToolTipText = "Edge Detection"
65 Set Picture0.Picture = picObject0

```

```

For X = 0 To intUpperBoundX - 1
  For Y = 0 To intUpperBoundY - 1

5    If (X = 0 Or X = intUpperBoundX - 1 Or Y = 0 Or Y = intUpperBoundY - 1) Then

        bytRed = ImagePixels(0, X, Y)
        bytBlue = ImagePixels(1, X, Y)
        bytGreen = ImagePixels(2, X, Y)
10    RGBLong = RGB(bytRed, bytGreen, bytBlue)

        Picture0.PSet (X, Y), RGBLong

    Else

15    G_X = 0
        G_Y = 0
        G_X_Y = 0

20    bRXY = ImagePixels(0, X, Y)
        bRXYp1 = ImagePixels(0, X, Y + 1)
        bRXm1Y = ImagePixels(0, X - 1, Y)
        bRXYm1 = ImagePixels(0, X, Y - 1)
        bRXm1Yp1 = ImagePixels(0, X - 1, Y + 1)
25    bRXm1Ym1 = ImagePixels(0, X - 1, Y - 1)
        bRXp1Y = ImagePixels(0, X + 1, Y)
        bRXp1Ym1 = ImagePixels(0, X + 1, Y - 1)
        bRXp1Yp1 = ImagePixels(0, X + 1, Y + 1)

30    G_X = bRXp1Ym1 + 2 * bRXp1Y + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXm1Y - bRXm1Yp1
        G_Y = bRXm1Yp1 + 2 * bRXYp1 + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXYm1 - bRXp1Ym1
        G_X_Y = Sqr((G_X * G_X) + (G_Y * G_Y))

        bytRed = G_X_Y

35    bRXY = ImagePixels(1, X, Y)
        bRXYp1 = ImagePixels(1, X, Y + 1)
        bRXm1Y = ImagePixels(1, X - 1, Y)
        bRXYm1 = ImagePixels(1, X, Y - 1)
40    bRXm1Yp1 = ImagePixels(1, X - 1, Y + 1)
        bRXm1Ym1 = ImagePixels(1, X - 1, Y - 1)
        bRXp1Y = ImagePixels(1, X + 1, Y)
        bRXp1Ym1 = ImagePixels(1, X + 1, Y - 1)
        bRXp1Yp1 = ImagePixels(1, X + 1, Y + 1)

45    G_X = bRXp1Ym1 + 2 * bRXp1Y + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXm1Y - bRXm1Yp1
        G_Y = bRXm1Yp1 + 2 * bRXYp1 + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXYm1 - bRXp1Ym1
        G_X_Y = Sqr((G_X * G_X) + (G_Y * G_Y))

50    bytBlue = G_X_Y

        bRXY = ImagePixels(2, X, Y)
        bRXYp1 = ImagePixels(2, X, Y + 1)
        bRXm1Y = ImagePixels(2, X - 1, Y)
55    bRXYm1 = ImagePixels(2, X, Y - 1)
        bRXm1Yp1 = ImagePixels(2, X - 1, Y + 1)
        bRXm1Ym1 = ImagePixels(2, X - 1, Y - 1)
        bRXp1Y = ImagePixels(2, X + 1, Y)
        bRXp1Ym1 = ImagePixels(2, X + 1, Y - 1)
60    bRXp1Yp1 = ImagePixels(2, X + 1, Y + 1)

        G_X = bRXp1Ym1 + 2 * bRXp1Y + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXm1Y - bRXm1Yp1
        G_Y = bRXm1Yp1 + 2 * bRXYp1 + bRXp1Yp1 - bRXm1Ym1 - 2 * bRXYm1 - bRXp1Ym1
        G_X_Y = Sqr((G_X * G_X) + (G_Y * G_Y))

65    bytGreen = G_X_Y

```

```
Picture0.PSet (X, Y), RGB(bytRed, bytGreen, bytBlue)
```

```
End If
```

5

```
Next Y
Next X
```

```
End Sub
```

10 Private Sub cmdGray\_Shade\_Click()

```
Set Picture0.Picture = picObject0
```

```
For X = 0 To intUpperBoundX - 1
```

```
For Y = 0 To intUpperBoundY - 1
```

15 Picture0.PSet (X, Y), Picture0.Point(X, Y) - 5

```
Next Y
```

```
Next X
```

```
MsgBox "Gray shades routine ended !"
```

20

```
End Sub
```

```
Private Sub cmdDis_and_Orientation_Click()
```

25 Const interval\_range = 7

```
Dim WhitePixel, BlackPixel As Long
```

```
Dim linescan As Integer
```

```
Dim i, j, k, l, IntX, Temp_X, Temp_Y As Integer
```

30 Dim Flag, SumTline, Dummy As Integer

```
Dim interval As Integer
```

```
Dim ZeroO_X, ZeroO_Y As Double
```

```
Dim L1SlopeR, L2SlopeR, L3SlopeR, L4SlopeR, L1SlopeY, _
L2SlopeY, L3SlopeY, L4SlopeY, AvgSlope As Double
```

35

```
Dim UpperBound, LowerBound As Double
```

```
Dim InterceptY As Integer
```

```
Dim Count_Points(0 To 403) As Integer
```

```
Dim TempInt, Choice As Integer
```

40

```
Dim Dum(0 To 15) As Double
```

```
Dim TempDouble As Double
```

```
Dim Tline(0 To 50) As Integer
```

```
Dim Oripixels(0 To intUpperBoundX, 0 To intUpperBoundY) As Long
```

45

```
Dim Displacement, Angle, Theta As Double
```

```
Dim CenterLineSlope As Double
```

```
Dim CenterLineIntercept As Double
```

```
Dim Center_Point_X, Center_Point_Y As Double
```

```
Dim TempText As String
```

50

```
cmdDis_and_Orientation.ToolTipText = "Displacement and Orientation"
```

```
Open "c:\windows\desktop\InspResults.txt" For Output As #1
```

55

```
For X = 0 To intUpperBoundX - 1
```

```
For Y = 0 To intUpperBoundY - 1
```

```
Oripixels(X, Y) = pixels(X, Y)
```

```
Next Y
```

```
Next X
```

60

```
'Apply the binary image technique
```

```
For X = 0 To intUpperBoundX - 1
```

```
For Y = 0 To intUpperBoundY - 1
```

65

```
If (Oripixels(X, Y) < RGB(255, 255, 255)) Then
```

```
Oripixels(X, Y) = 0
```

```

Else
  Oripixels(X, Y) = RGB(255, 255, 255)
End If
Picture0.PSet (X, Y), Oripixels(X, Y)
5
Next Y
Next X

Find the number of center lines
10 'A line is BW...WB; if there is less than four BW...WBs; then Image is tiled
'white interval should be less than 7 for the central lines
'use Black/White/Black to find a line

linescan = 0
15 interval = 1

For Y = 50 To intUpperBoundY - 1
  Tline(linescan) = 0
  Flag = 0
20 l = 0

  For X = 0 To intUpperBoundX - 1
    If ((Oripixels(X, Y) = RGB(0, 0, 0)) And _
      (Oripixels(X + 1, Y) = RGB(255, 255, 255))) Then
25
      For interval = 1 To interval_range - 1
        If (Oripixels(X + 1 + interval, Y) = RGB(0, 0, 0)) Then

          Tline(linescan) = Tline(linescan) + 1
          Flag = 1
30
          Coord_X(linescan, l) = X + 1      'of each line
          Coord_Y(linescan, l) = Y

35          l = l + 1

        End If
        interval = interval_range 'stop the for loop
        Next interval
      End If
      Next X
      Y = Y + 10 ' 5      'to have 40 arbitrary verticle lines
      If (Flag = 1) Then
        linescan = linescan + 1
45      End If
      Next Y

      k = 0
      SumTline = 0
50      For j = 0 To linescan - 1 'from prev. routine # of arb. ver. lines
        If (Tline(j) > 0) Then
          SumTline = SumTline + Tline(j)
          k = k + 1
        End If
55      Next j

      If (3.5 <= (SumTline / k) <= 4.5) Then
        MsgBox ("Number of center lines is " & n)

60      L1SlopeR = GetSlope(linescan, 0, 0)
      L1SlopeY = GetSlope(linescan, 0, 1)

      L2SlopeR = GetSlope(linescan, 1, 0)
      L2SlopeY = GetSlope(linescan, 1, 1)
65
      L3SlopeR = GetSlope(linescan, 2, 0)

```

L3SlopeY = GetSlope(linescan, 2, 1)

L4SlopeR = GetSlope(linescan, 3, 0)

L4SlopeY = GetSlope(linescan, 3, 1)

5

AvgSlope = (L1SlopeY + L2SlopeY + L3SlopeY + L4SlopeY) / 4

LowerBound = 0.025 \* AvgSlope

UpperBound = 1.025 \* AvgSlope

10

'Use the absolute value; therefore, it works on both +/- values

If ((Abs(LowerBound) <= Abs(L1SlopeY) <= Abs(UpperBound)) And \_

(Abs(LowerBound) <= Abs(L2SlopeY) <= Abs(UpperBound)) And \_

(Abs(LowerBound) <= Abs(L3SlopeY) <= Abs(UpperBound)) And \_

15

(Abs(LowerBound) <= Abs(L4SlopeY) <= Abs(UpperBound))) Then

MsgBox ("Four lines are parallel !")

Else: MsgBox ("Potential errors in finding parallel lines")

End If

20

Else

MsgBox ("Number of center lines is " & SumTline / k)

End If

The following is to find the center point of the image

25

'Step 1: Find the black pixel

'Step 2: Calculate the neighborhood pixels distance to the regression line

'Step 3: Locate the one that has the smallest distance

'Step 4: Check to see if the feature of w

30

'

www

'been meet

W

'if not; based on current X, Y; go to Step 1

35

BlackPixel = RGB(0, 0, 0)

WhitePixel = RGB(255, 255, 255)

CenterLineSlope = GetSlope(linescan, 0, 2)

CenterLineIntercept = GetSlope(linescan, 0, 3)

40

MsgBox ("C.L.Intercept = " & CenterLineIntercept)

MsgBox ("C.L.Slope = " & CenterLineSlope)

For Y = 20 To intUpperBoundY - 1

45

X = (Y \* CenterLineSlope) + CenterLineIntercept

IntX = X

If (Oripixels(IntX, Y) = BlackPixel) Then

l = 0

50

Temp\_X = 0

Temp\_Y = 0

For i = -1 To 1

For j = -1 To 1

If (Oripixels(IntX + i, Y + j) = WhitePixel) Then

55

Temp\_X = Temp\_X + (IntX + i)

Temp\_Y = Temp\_Y + (Y + j)

l = l + 1

End If

If (l >= 3) Then Neighborhood pixels are White

60

Center\_Point\_X = Temp\_X / l

Center\_Point\_Y = Temp\_Y / l

MsgBox ("Center X = " & Center\_Point\_X)

Beep

MsgBox ("Center Y = " & Center\_Point\_Y)

65

i = 1

j = 1

Y = intUpperBoundY

```

        End If
    Next j
Next i

5   l = 0
    Dum(l) = 0
    For i = 0 To 1
        For j = 0 To 1
            If (i <> 0 Or j <> 0) Then
10      Dum(l) = Measure_Distance(CenterLineIntercept, CenterLineSlope, X + i, Y + j)
                l = l + 1
            End If
        Next j
    Next i

15  For k = 0 To l - 1
        If (Dum(k) < Dum(k + 1)) Then
            TempDouble = Dum(k)
            Dum(k) = Dum(k + 1)
            Dum(k + 1) = TempDouble
20      End If
    Next k

    For i = 0 To 1
        For j = 0 To 1
            If ((i <> 0 Or j <> 0) And (Dum(l - 1) = Measure_Distance(CenterLineIntercept, CenterLineSlope, X + i,
25  Y + j))) _
                Then _
30      X = X + i
                Y = Y + j - 1 'because Y auto. inc. by 1
                i = 1
                j = 1

35      End If
        Next j
    Next i

    End If
40  Picture0.PSet (IntX, Y), RGB(255, 255, 255)

Next Y

The following section is to find the orientation and displacement
45 'Comparing the theoretical zero point and new zero point
'Calculate the displacement and titled angle

ZeroO_X = (intUpperBoundX - 1) / 2
ZeroO_Y = (intUpperBoundY - 1) / 2
50 If ((Center_Point_X - ZeroO_X = 0) And (Center_Point_Y - ZeroO_Y = 0)) Then

    Theta = 0
    Displacement = 0
55 Else

    Displacement = Sqr((Center_Point_X - ZeroO_X) ^ 2 + (Center_Point_Y - ZeroO_Y) ^ 2)
    TempDouble = (Center_Point_Y - ZeroO_Y) / Displacement
60  Theta = Atn(TempDouble / Sqr(-TempDouble * TempDouble + 1))
    Angle = 90 - ((Theta / 3.141592654) * 180)

End If

65 MsgBox ("Titled angle is (clockwise): " & Angle)
   MsgBox ("Displacement is: " & Displacement)

```

```

For X = 0 To intUpperBoundX - 1
For Y = 0 To intUpperBoundY - 1
    Picture0.PSet (X, Y), RGB(255, 255, 255)
5   Next Y
    Next X

For i = 0 To 6

10   Picture0.CurrentX = 20
    Picture0.CurrentY = 20 + 15 * i
    Select Case i
        Case 0:
15       Picture0.Print ("Number of center lines are " & n)
            TempText = "Number of center lines are: "
            Write #1, TempText, n

        Case 1:
20       Picture0.Print ("C.L.Intercept = " & CenterLineIntercept)
            Write #1, "C.L.Intercept = ", CenterLineIntercept

        Case 2:
25       Picture0.Print ("C.L.Slope = " & CenterLineSlope)
            Write #1, "C.L.Slope = ", CenterLineSlope

        Case 3:
30       Picture0.Print ("Center X = " & Center_Point_X)
            Write #1, "Center X = ", Center_Point_X

        Case 4:
35       Picture0.Print ("Center Y = " & Center_Point_Y)
            Write #1, "Center Y = ", Center_Point_Y

        Case 5:
40       Picture0.Print ("Titled angle is (clockwise): " & Angle)
            Write #1, "Titled angle is (clockwise): ", Angle

        Case 6:
45       Picture0.Print ("Displacement is: " & Displacement)
            Write #1, "Displacement is ", Displacement

    End Select
Next i
Close #1

45   MsgBox "Ori & Dis. routine ended !"

End Sub

50   Private Sub cmdQuit_Click()

    cmdQUIT.ToolTipText = "Exit the system"

55       Unload Form4
        Exit Sub
        'Form2.Show

End Sub

60   Private Sub cmdBack_Click()

    cmdBack.ToolTipText = "Back to previous stage"

    Unload Form4
    Image_Capture.Show

65

```



```

End Sub
Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub
5 Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub
Function GetRed(colorVal As Long) As Integer
    GetRed = colorVal Mod 256
10 End Function
Function GetGreen(colorVal As Long) As Integer
    GetGreen = ((colorVal And &HFF00FF00) / 256&)
End Function
Function GetBlue(colorVal As Long) As Integer
15 GetBlue = (colorVal And &HFF0000) / (256& * 256&)
End Function
Function GetSlope(Points As Integer, LineN As Integer, Choice As Integer) As Double

    Dim SumXY, SumX, SumY As Double
    Dim SumYsq, SumXsq, FuncDummy As Double
    Dim a, Index, Position_X, Position_Y As Integer

    SumXY = 0
    SumX = 0
25 SumY = 0
    SumXsq = 0
    SumYsq = 0
    Position_X = 0
    Position_Y = 0
30 Index = 0
    FuncDummy = 0

    'Sometimes the image is truncated; u do not have
    'all the 18 points; we use the B to represent to count
    'all the points
35 'Choice 0: Line correlation coefficient
    'Choice 1: Parallel line slope
    'Choice 2: Center line slope
    'Choice 3: Center line intercept
40

    If (Choice = 0 Or Choice = 1) Then
        For a = 0 To Points - 1
            Position_X = Coord_X(a, LineN)
            Position_Y = Coord_Y(a, LineN)
45

            If ((Position_X <> 0) And (Position_Y <> 0)) Then
                SumXY = SumXY + (Position_X * Position_Y)
                SumX = SumX + Position_X
                SumY = SumY + Position_Y
50                SumYsq = SumYsq + Position_Y ^ 2
                SumXsq = SumXsq + Position_X ^ 2
                Index = Index + 1
            End If
        Next a
55

    End If

    If (Choice = 2 Or Choice = 3) Then
        For a = 0 To Points - 1
            For LineN = 0 To n - 1
60                Position_X = Coord_X(a, LineN)
                Position_Y = Coord_Y(a, LineN)

                If ((Position_X <> 0) And (Position_Y <> 0)) Then
65                    SumXY = SumXY + (Position_X * Position_Y)
                    SumX = SumX + Position_X

```

```

SumY = SumY + Position_Y
SumYsq = SumYsq + Position_Y ^ 2
SumXsq = SumXsq + Position_X ^ 2
Index = Index + 1

```

5

```

End If
Next LineN
Next a
End If

```

10

```

If ((SumX = 0) Or (SumY = 0) Or (SumX * SumY = 0)) Then
    GetSlope = 0

```

```
Else

```

15

```

    If (Choice = 1 Or Choice = 2) Then
        GetSlope = ((SumXY) - ((SumX * SumY) / Index)) / ((SumYsq) - ((SumY * SumY) / Index))
    End If

```

```

    If (Choice = 3) Then

```

20

```

        FuncDummy = ((SumXY) - ((SumX * SumY) / Index)) / ((SumYsq) - ((SumY * SumY) / Index))
        GetSlope = (SumX - (FuncDummy * SumY)) / Index
    End If

```

```

    If (Choice = 0) Then

```

25

```

        FuncDummy = Sqr((SumXsq - (SumX ^ 2 / Index)) * (SumYsq - (SumY ^ 2 / Index)))
        GetSlope = ((SumXY) - ((SumX * SumY) / Index)) / FuncDummy

```

```

    End If
End If

```

30

```

End Function
Function dblSquare(SquareMe As Integer) As Double

```

```

    dblSquare = SquareMe ^ 2 * SquareMe

```

35

```

End Function
Function Measure_Distance(c1 As Double, m1 As Double, Point2_X As Integer, Point2_Y As Integer) As Double

```

```

    Dim Point1_X, Point1_Y As Long
    Dim c2 As Long

```

40

```

    c2 = Point2_X - ((-1 / m1) * Point2_Y)
    Point1_X = (c2 * m1 - c1 * (-1 / m1)) / (m1 - (-1 / m1))
    Point1_Y = (c2 - c1) / (m1 - (-1 / m1))
    Measure_Distance = Sqr((Point2_X - Point1_X) ^ 2 + (Point2_Y - Point1_Y) ^ 2)

```

45

```

End Function

```

```

Private Sub Picture0_Click()

```

50

```

End Sub

```

---

### **820F5 FORM**

```

Private Sub cmdBack_Click()

```

55

```

    cmdBack.ToolTipText = "Back to previous stage"

```

```

Unload Form5
Form4.Show

```

60

```

End Sub

```

```

Private Sub Picture2_Click()

```

```

End Sub

Private Sub cmdQuit_Click()
5  cmdQuit.ToolTipText = "Exit from the system"

  Unload Form5
  Exit Sub

10 End Sub
Private Sub cmdShowRes_Click()

  Dim NewLine As String

15  cmdShowRes.ToolTipText = "Display the inspection results"

  On Error GoTo FileError
  Open "c:\windows\desktop\InspResults.txt" For Input As #1
  Do Until EOF(1)
20    Line Input #1, NewLine
    TEXT1.Text = TEXT1.Text + NewLine + vbCrLf
  Loop

  Exit Sub

25 FileError:
  MsgBox "File Error! "

  End Sub

30 Private Sub Form_Load()

  End Sub

```

### 35 IMAGE CAPTURE FORM

```

Dim cfg As VPX_Config
Dim hDIB As Long
Dim numAverage As Integer
40 Dim prevAverageIndex As Integer
Dim filtOn(8) As Boolean
Dim filt(8) As Long
Dim avgNums(6) As Integer
Dim Err As Integer

45 Private Declare Function GlobalFree Lib "KERNEL32" (ByVal handle&) As Long
Sub SetupMenu()
  Dim Enable As Boolean
  Enable = cfg.outputFormat = VPP_mono Or cfg.outputFormat = VPP_BGR24

50  ImageCaptureFunctionEnable initialized in the password Form

  For i% = 1 To 6
    If (ImageCaptureFunctionsEnable = 0) Then
55      Average(i%).Enabled = False
    End If
    If (ImageCaptureFunctionsEnable = 1) Then
      Average(i%).Enabled = True
    End If
  Next i%

60  For i% = 1 To 8
    If (ImageCaptureFunctionsEnable = 0) Then
      Filter(i%).Enabled = False
    End If
  Next i%

```

65

65

The following is the boundary of the image

5 Line (220, 7550)-(11400, 7550), RGB(255, 255, 0) Bot.Hoz.  
 Line (220, 280)-(11400, 280), RGB(255, 255, 0) Top Hoz.  
 Line (220, 7550)-(220, 280), RGB(255, 255, 0) Left Ver.  
 Line (11400, 7550)-(11400, 280), RGB(255, 255, 0) Rgt Ver.

The following is the cross-hair of the area

10 Line (5675, 3913)-(5975, 3913), RGB(255, 255, 0) Horizontal line  
 Line (5825, 4013)-(5825, 3813), RGB(255, 255, 0) Verticle line

15 Timer1.Enabled = True  
 End Sub

---

### **820PASSWORD FORM**

Private Sub cmdLogin\_Click()

20 If txtpasswd.Text = "password" Then  
 Unload password

Initialize the settigns

25 DummyY = 0  
 ImageCaptureFunctionsEnable = 0

Form2.Show  
 MsgBox "Please set up the HMD first !"

30 Else  
 MsgBox "Wrong Passord ! Please enter again !"

35 End If

End Sub

40 Private Sub cmdRestart\_Click()  
 txtpasswd.Text = ""  
 End Sub

45 Private Sub cmdQuit\_Click()  
 Unload password  
 exitwnd.Show  
 End Sub

50 Private Sub Form\_Load()  
 txtpasswd.Text = ""  
 End Sub

Private Sub Image2\_Click()

End Sub

---

### **55 PASSWORDFORSETTINGS FORM**

Private Sub Command1\_Click()  
 If txtpasswd.Text = "passwordforsettings" Then  
 specs.Show  
 specs!TEXT1.Text = ""  
 specs!Text2.Text = ""  
 specs!Text3.Text = ""

```

specs!Text4.Text = ""
Unload passwordforsettings
Else
5   MsgBox "Wrong Passard ! Please enter again !"

End If
End Sub

```

```

10  Private Sub Command2_Click()
    txtpasswd.Text = ""
End Sub

```

```

Private Sub Command3_Click()

```

```

15  Unload passwordforsettings
    exitwnd.Show

End Sub

```

```

20  Private Sub Form_Load()
    txtpasswd.Text = ""
End Sub

```

```

25  Private Sub Image2_Click()

End Sub

```

---

#### SETTINGS FORM

```

30  Private Sub end_Click()

    Unload Me
    Form2.Show

```

```

35  End Sub

Private Sub Image2_Click()

End Sub

```

```

40  Private Sub Reset_Click()

    Unload Me
    specs.Show

```

```

45  End Sub

```

---

#### SPECTEST FORM

```

50  Private Sub Command1_Click()

```

```

    settings.Show
    settings!Text1.Text = specs!Text1.Text 'Height
55  settings!Text2.Text = specs!Text2.Text 'W2
    settings!Text3.Text = specs!Text3.Text 'Width
    settings!Text4.Text = specs!Text4.Text 'W1

```

```

60  'The following are the Public variables
    'Declared in the Image_Capture_Module
    'Val() convert the string into integer

```

```

PatternWidth = Val(Text3.Text)
PatternHeight = Val(Text1.Text)
PatternW1 = Val(Text4.Text)
PatternW2 = Val(Text2.Text)

```

5

```

The following are the testing routines
Height = Val(Text1.Text)
MsgBox ("Height is " & Str(PatternHeight))
MsgBox ("Width is " & Str(PatternWidth))
MsgBox ("W1 is " & Str(PatternW1))
MsgBox ("W2 is " & Str(PatternW2))

```

10

```

Unload specs
End Sub

```

15

```
Private Sub Disable_Click()
```

```
ImageCaptureFunctionsEnable = 0
```

20

```
End Sub
```

```
Private Sub Enable_Click()
```

```
ImageCaptureFunctionsEnable = 1
```

25

```
End Sub
```

```
Private Sub Image2_Click()
```

30

```
End Sub
```

---

### IMAGE CAPTURE MODULE

```

Public ImageCaptureFunctionsEnable As Integer
Public PatternWidth, PatternHeight, PatternW1, PatternW2 As Integer

```

35

```

Public DummyY As Integer
Public FlagLabel As Integer

```

40

```

'Bool
Global Const VPP_false = 0
Global Const VPP_true = 1

```

```
'Error numbers
```

45

```

Global Const VPP_success = 0          No error
Global Const VPP_toolkitInUse = 1     'VideoPort toolkit is already in use
Global Const VPP_noHardwareDetected = 2   No VideoPort hardware detected
Global Const VPP_noDriverDetected = 3     No VideoPort PCMCIA driver detected
Global Const VPP_oldVideoPortDetected = 4   The installed VideoPort is old-style
Global Const VPP_notInitialized = 5       'init has not been called
Global Const VPP_notConfigured = 6        'videoConfig has not been called
Global Const VPP_snapNotPrepared = 7      'prepareSnap has not been called
Global Const VPP_snapNotStarted = 8       'startSnap has not been called
Global Const VPP_snapNotFinished = 9      'finishSnap has not been called
Global Const VPP_readoutNotStarted = 10    'startReadout has not been called
Global Const VPP_noSignalDetected = 11     No video signal detected
Global Const VPP_noColorSnapped = 12      Snapped image does not contain colour
Global Const VPP_readoutOutsideSnappedImage = 13 'Attempt to read outsize snapped image
Global Const VPP_parameterOutOfRange = 14   Parameter to function is out of range
Global Const VPP_imageWidthOutOfRange = 15   Image width is out of range
Global Const VPP_imageHeightOutOfRange = 16   Image height is out of range
Global Const VPP_badPointer = 17           Bad pointer (possibly NULL)
Global Const VPP_lostContact = 18          Contact with VideoPort is lost
Global Const VPP_outOfMemory = 19          Could not claim the memory needed

```

50

55

60



Global Const VPP\_fileIOError = 20 File input/output error

Global constants

Global Const VPP\_DEFAULT\_CHANNEL = 0

5 Global Const VPP\_MIN\_BRIGHTNESS = -128

Global Const VPP\_MIN\_CONTRAST = -128

Global Const VPP\_MIN\_SATURATION = -128

Global Const VPP\_MIN\_HUE = -128

Global Const VPP\_DEFAULT\_BRIGHTNESS = 0

10 Global Const VPP\_DEFAULT\_CONTRAST = 0

Global Const VPP\_DEFAULT\_SATURATION = 0

Global Const VPP\_DEFAULT\_HUE = 0

Global Const VPP\_DEFAULT\_GAMMA = 1

Global Const VPP\_MAX\_BRIGHTNESS = 127

15 Global Const VPP\_MAX\_CONTRAST = 127

Global Const VPP\_MAX\_SATURATION = 127

Global Const VPP\_MAX\_HUE = 127

Global Const VPP\_DEFAULT\_FLASH\_DELAY = 8

20 Video standards

Global Const VPP\_NTSC = 0

Global Const VPP\_PAL = 1

Global Const VPP\_noSignal = 2

25 Signal types

Global Const VPP\_composite = 0

Global Const VPP\_Svideo = 1

Global Const VPP\_monochrome = 2

30 Type VPP\_SnapData

xOffset As Integer X offset of active video area

xActive As Integer Width of active video area

xPixels As Integer Requested width in pixels of active video area

yOffset As Integer Y offset of active video area

35 yActive As Integer Height of active video area

yPixels As Integer Requested height in pixels of active video area

interlaced As Long Flag to turn on interlaced snap

monochrome As Long Flag to turn on monochrome snap

End Type

40

Type VPP\_LimitData

xActiveMax As Integer PAL: 922, NTSC: 754

xPixelsMax As Integer PAL: 922, NTSC: 754, VideoPort Junior: 510

xActiveRatio As Integer Currently: 14

45 xPixelsRatio As Integer Currently: 1

yActiveMax As Integer PAL: 610, NTSC: 510

yPixelsMax As Integer PAL: 610, NTSC: 510, VideoPort Junior: 510

yActiveRatio As Integer Currently: 14

yPixelsRatio As Integer Currently: 1

50 End Type

Readout modes Format: Size factor:

Global Const VPP\_mono = 1 Byte I... 1

Global Const VPP\_mono4 = 2 Nibble IIII(2)... dithered 1/2

55 Global Const VPP\_RGB8 = 3 Byte RRRGGGGBB(2)... dithered 1

Global Const VPP\_RGB15 = 4 Word 0RRRRRRGG GGGBBBBB(2)... 2

Global Const VPP\_RGB16 = 5 Word RRRRRRRGGG GGGBBBBB(2)... 2

Global Const VPP\_BGR24 = 7 Byte B,G,R... 3

Global Const VPP\_BGR032 = 8 Byte B,G,R,0... 4

60

Declare Function VPP\_init Lib "VPX32.DLL" () As Long

Declare Function VPP\_closedown Lib "VPX32.DLL" (ByVal powerOff&) As Long

Declare Function VPP\_getCurrentCardHandle Lib "VPX32.DLL" (cardhandle&) As Long

Declare Function VPP\_setActiveCard Lib "VPX32.DLL" (ByVal cardhandle&) As Long

65 Declare Function VPP\_videoConfig Lib "VPX32.DLL" (ByVal channel%, ByVal signalType&, videoStandard&) As Long

Declare Function VPP\_testSignal Lib "VPX32.DLL" (videoStandard&) As Long  
 Declare Function VPP\_setBrightness Lib "VPX32.DLL" (ByVal brightness%) As Long  
 Declare Function VPP\_setContrast Lib "VPX32.DLL" (ByVal contrast%) As Long  
 Declare Function VPP\_setSaturation Lib "VPX32.DLL" (ByVal saturation%) As Long  
 5 Declare Function VPP\_setHue Lib "VPX32.DLL" (ByVal hue%) As Long  
 Declare Function VPP\_setGamma Lib "VPX32.DLL" (ByVal gamma!) As Long  
 Declare Function VPP\_enableFlash Lib "VPX32.DLL" (ByVal flashSelect&, ByVal mustBeNULL&, ByVal  
 flashDelay%) As Long  
 Declare Function VPP\_disableFlash Lib "VPX32.DLL" () As Long  
 10 Declare Function VPP\_getLimits Lib "VPX32.DLL" (ByVal videoStandard&, limitData As VPP\_LimitData) As  
 Long  
 Declare Function VPP\_prepareSnap Lib "VPX32.DLL" (snapData As VPP\_SnapData) As Long  
 Declare Function VPP\_startSnap Lib "VPX32.DLL" () As Long  
 Declare Function VPP\_finishSnap Lib "VPX32.DLL" () As Long  
 15 Declare Function VPP\_flashSnap Lib "VPX32.DLL" () As Long  
 Declare Function VPP\_extTrigSnap Lib "VPX32.DLL" (ByVal msecTimeout%) As Long  
 Declare Function VPP\_autoCrop Lib "VPX32.DLL" (snapData As VPP\_SnapData) As Long  
  
 Type VPX\_Config  
 20 videoStandard As Long  
 signalType As Long  
 inputChannel As Long  
 brightness As Long  
 contrast As Long  
 25 saturation As Long  
 gamma As Single  
 hue As Long  
 snapDataNTSC As VPP\_SnapData  
 snapDataPAL As VPP\_SnapData  
 30 outputFormat As Long  
 extTrigSnap As Long  
 flashSnap As Long  
 extTrigEnable As Long  
 flashEnable As Long  
 35 mono4Enable As Long  
 monoEnable As Long  
 RGB8Enable As Long  
 BGR24Enable As Long  
 RGB15Enable As Long  
 40 RGB16Enable As Long  
 BGR032Enable As Long  
 End Type  
  
 Type VPX\_Filter  
 45 Width As Byte  
 Height As Byte  
 divideBy As Long  
 doAbs As Long  
 The following scheme allows up to 4x4 or 3x5 filters  
 50 coeff0 As Byte      Treat with care, should be "signed char"  
 coeff1 As Byte      Treat with care, should be "signed char"  
 coeff2 As Byte      Treat with care, should be "signed char"  
 coeff3 As Byte      Treat with care, should be "signed char"  
 coeff4 As Byte      Treat with care, should be "signed char"  
 55 coeff5 As Byte      Treat with care, should be "signed char"  
 coeff6 As Byte      Treat with care, should be "signed char"  
 coeff7 As Byte      Treat with care, should be "signed char"  
 coeff8 As Byte      Treat with care, should be "signed char"  
 coeff9 As Byte      Treat with care, should be "signed char"  
 60 coeff10 As Byte      Treat with care, should be "signed char"  
 coeff11 As Byte      Treat with care, should be "signed char"  
 coeff12 As Byte      Treat with care, should be "signed char"  
 coeff13 As Byte      Treat with care, should be "signed char"  
 coeff14 As Byte      Treat with care, should be "signed char"  
 65 coeff15 As Byte      Treat with care, should be "signed char"  
 End Type

Predefined filters

Global Const VPX\_AVERAGE = 1

Global Const VPX\_SMOOTH = 2

5 Global Const VPX\_DETAIL = 3

Global Const VPX\_SHARPEN = 4

Global Const VPX\_AI = 5

Global Const VPX\_AISHARPEN = 6

Global Const VPX\_VERTLINES = 7

10 Global Const VPX\_HORIZLINES = 8

Declare Function VPX\_defaultConfig Lib "VPX32.DLL" (config As VPX\_Config) As Long

Declare Function VPX\_prepare Lib "VPX32.DLL" (config As VPX\_Config, ByVal forceConfig&) As Long

Declare Function VPX\_snap Lib "VPX32.DLL" (config As VPX\_Config) As Long

15 Declare Function VPX\_draw Lib "VPX32.DLL" (ByVal hDC&, config As VPX\_Config, ByVal X%, ByVal Y%,  
ByVal x1%, ByVal y2%) As Long

Declare Function VPX\_drawDIB Lib "VPX32.DLL" (ByVal hDC&, ByVal hDIB&, ByVal X%, ByVal Y%,  
ByVal x1%, ByVal y2%) As Long

Declare Function VPX\_readoutDIB Lib "VPX32.DLL" (config As VPX\_Config, hDIB&) As Long

20 Declare Function VPX\_readoutDIBToClipboard Lib "VPX32.DLL" (config As VPX\_Config) As Long

Declare Function VPX\_saveDIBToClipboard Lib "VPX32.DLL" (ByVal hDIB&) As Long

Declare Function VPX\_readoutDIBToFile Lib "VPX32.DLL" (config As VPX\_Config, ByVal filename\$) As  
Long

Declare Function VPX\_saveDIBToFile Lib "VPX32.DLL" (ByVal hDIB&, ByVal filename\$) As Long

25 Declare Function VPX\_snapAverageDIB Lib "VPX32.DLL" (config As VPX\_Config, hDIB&, ByVal  
numAverage%, ByVal divideBy%) As Long

Declare Function VPX\_filterDIB Lib "VPX32.DLL" (ByVal hDIB&, Filter As VPX\_Filter) As Long

Declare Function VPX\_filterDIBPredef Lib "VPX32.DLL" Alias "VPX\_filterDIB" (ByVal hDIB&, ByVal  
Filter&) As Long

30 Declare Function VPX\_getFilter Lib "VPX32.DLL" (ByVal filterNo&) As Long

Declare Function VPX\_copyDIB Lib "VPX32.DLL" (ByVal hDIB&, phDIB&) As Long

Declare Function VPX\_readIniFile Lib "VPX32.DLL" (ByVal File\$, ByVal section\$, config As VPX\_Config) As  
Long

35 Declare Function VPX\_saveIniFile Lib "VPX32.DLL" (ByVal File\$, ByVal section\$, config As VPX\_Config) As  
Long

Declare Function VPX\_defaultPalette Lib "VPX32.DLL" (ByVal hDC&, ByVal monochrome&) As Long

Declare Function VPX\_releasePalette Lib "VPX32.DLL" () As Long

Declare Function VPX\_formatDialogBox Lib "VPX32.DLL" (ByVal ignored&, ByVal parent&, config As  
VPX\_Config) As Long

40 Sub Check(ret As Integer)

If ret <> VPP\_success Then

MsgBox "Error returned from VPPTOOLS: " + Chr\$(10) + VPP\_errorString(ret), MB\_OK +

MB\_ICONSTOP, "Test application"

45 ret = VPP\_closedown(VPP\_true)

End

End If

End Sub

50 Function VPP\_errorString(errno As Integer) As String

Select Case errno

Case VPP\_success

VPP\_errorString = "No error"

Case VPP\_toolkitInUse

55 VPP\_errorString = "VideoPort toolkit is already in use"

Case VPP\_noHardwareDetected

VPP\_errorString = "No VideoPort hardware detected"

Case VPP\_noDriverDetected

VPP\_errorString = "No VideoPort PCMCIA driver detected"

60 Case VPP\_oldVideoPortDetected

VPP\_errorString = "The installed VideoPort is old-style"

Case VPP\_notInitialized

VPP\_errorString = "init has not been called"

Case VPP\_notConfigured

65 VPP\_errorString = "videoConfig has not been called"

Case VPP\_snapNotPrepared

VPP\_errorString = "prepareSnap has not been called"  
 Case VPP\_snapNotStarted  
 VPP\_errorString = "startSnap has not been called"  
 Case VPP\_snapNotFinished  
 VPP\_errorString = "finishSnap has not been called"  
 Case VPP\_readoutNotStarted  
 VPP\_errorString = "startReadout has not been called"  
 Case VPP\_noSignalDetected  
 VPP\_errorString = "No video signal detected"  
 Case VPP\_noColorSnapped  
 VPP\_errorString = "Snapped image does not contain colour"  
 Case VPP\_readoutOutsideSnappedImage  
 VPP\_errorString = "Attempt to read outside snapped image"  
 Case VPP\_parameterOutOfRange  
 VPP\_errorString = "Parameter to function is out of range"  
 Case VPP\_imageWidthOutOfRange  
 VPP\_errorString = "Image width is out of range"  
 Case VPP\_imageHeightOutOfRange  
 VPP\_errorString = "Image height is out of range"  
 Case VPP\_badPointer  
 VPP\_errorString = "Bad pointer (possibly NULL)"  
 Case VPP\_lostContact  
 VPP\_errorString = "Contact with VideoPort is lost"  
 Case VPP\_outOfMemory  
 VPP\_errorString = "Out of memory"  
 Case VPP\_fileIOError  
 VPP\_errorString = "File I/O error"  
 Case Else  
 VPP\_errorString = "Unknown error"  
 End Select  
 End Function  
 Public Function Mean(X As Integer, Y As Integer)

End Function  
 Public Function Center\_Point(X As Integer, Y As Integer)

End Function

# NIDAQ32 MODULE

\*\*\*\*\*  
 Declare Function AI\_Change\_Parameter% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function AI\_Check% Lib "nidaq32.dll" (ByVal a%, b%, c%)  
 Declare Function AI\_Clear% Lib "nidaq32.dll" (ByVal a%)  
 Declare Function AI\_Configure% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,  
 ByVal f%)  
 Declare Function AI\_Mux\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%)  
 Declare Function AI\_Read% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d%)  
 Declare Function AI\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function AI\_VRead% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d#)  
 Declare Function AI\_VScale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e#, ByVal  
 f%, g#)  
 Declare Function Align\_DMA\_Buffer% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c As Any, ByVal d%, ByVal  
 e%, f%)  
 Declare Function AO\_Calibrate% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function AO\_Configure% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e#,  
 ByVal f%)  
 Declare Function AO\_Change\_Parameter% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function AO\_Update% Lib "nidaq32.dll" (ByVal a%)

09703426 " 107100  
00209 924266

Declare Function AO\_VWrite% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c#)  
 Declare Function AO\_Write% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function Calibrate\_E\_Series% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d#)  
 Declare Function Calibrate\_59xx% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c#)  
 5 Declare Function Calibrate\_DSA% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c#)  
 Declare Function Config\_Alarm\_Deadband% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c\$, ByVal d#,  
 ByVal e#, ByVal f%, ByVal g%, ByVal h%, ByVal i&)  
 Declare Function Config\_ATrig\_Event\_Message% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c\$, ByVal d#,  
 ByVal e#, ByVal f%, ByVal g&, ByVal h&, ByVal i&, ByVal j%, ByVal k%, ByVal l&)  
 10 Declare Function Config\_DAQ\_Event\_Message% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c\$, ByVal  
 d%, ByVal e&, ByVal f&, ByVal g&, ByVal h&, ByVal i&, ByVal j%, ByVal k%, ByVal l&)  
 Declare Function Configure\_HW\_Analog\_Trigger% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal  
 d&, ByVal e&, ByVal f&)  
 15 Declare Function CTR\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,  
 ByVal f%)  
 Declare Function CTR\_EvCount% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function CTR\_EvRead% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%)  
 Declare Function CTR\_FOUT\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal  
 e%)  
 20 Declare Function CTR\_Period% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function CTR\_Pulse% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)  
 Declare Function CTR\_Rate% Lib "nidaq32.dll" (ByVal a#, ByVal b#, c%, d%, e%)  
 Declare Function CTR\_Reset% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function CTR\_Restart% Lib "nidaq32.dll" (ByVal a%, ByVal b%)  
 25 Declare Function CTR\_Simul\_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, ByVal d%)  
 Declare Function CTR\_Square% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)  
 Declare Function CTR\_State% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 Declare Function CTR\_Stop% Lib "nidaq32.dll" (ByVal a%, ByVal b%)  
 30 Declare Function DAQ\_Check% Lib "nidaq32.dll" (ByVal a%, b%, c&)  
 Declare Function DAQ\_Clear% Lib "nidaq32.dll" (ByVal a%)  
 Declare Function DAQ\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function DAQ\_DB\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%)  
 Declare Function DAQ\_DB\_HalfReady% Lib "nidaq32.dll" (ByVal a%, b%, c%)  
 35 Declare Function DAQ\_DB\_Transfer% Lib "nidaq32.dll" (ByVal a%, b As Any, c&, d%)  
 Declare Function DAQ\_Monitor% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d&, e As Any, f&,  
 g%)  
 Declare Function DAQ\_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d As Any, ByVal e&, ByVal  
 f#)  
 40 Declare Function DAQ\_Rate% Lib "nidaq32.dll" (ByVal a#, ByVal b%, c%, d%)  
 Declare Function DAQ\_Start% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d As Any, ByVal e&, ByVal  
 f%, ByVal g%)  
 Declare Function DAQ\_StopTrigger\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&)  
 Declare Function DAQ\_to\_Disk% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d\$, ByVal e&,  
 ByVal f#, ByVal g%)  
 45 Declare Function DAQ\_VScale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e#,  
 ByVal f&, g%, h#)  
 Declare Function DIG\_Block\_Check% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c&)  
 Declare Function DIG\_Block\_Clear% Lib "nidaq32.dll" (ByVal a%, ByVal b%)  
 50 Declare Function DIG\_Block\_In% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c As Any, ByVal d&)  
 Declare Function DIG\_Block\_Out% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c As Any, ByVal d&)  
 Declare Function DIG\_Block\_PG\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%,  
 ByVal e%, ByVal f%, ByVal g%)  
 Declare Function DIG\_DB\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)  
 55 Declare Function DIG\_DB\_HalfReady% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 Declare Function DIG\_DB\_Transfer% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c As Any, ByVal d&)  
 Declare Function DIG\_Grp\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)  
 Declare Function DIG\_Grp\_Mode% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,  
 ByVal f%, ByVal g%)  
 60 Declare Function DIG\_Grp\_Status% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 Declare Function DIG\_In\_Grp% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 Declare Function DIG\_In\_Line% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d%)  
 Declare Function DIG\_In\_Port% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 65 Declare Function DIG\_Line\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function DIG\_Out\_Grp% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function DIG\_Out\_Line% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function DIG\_Out\_Port% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)

Declare Function DIG\_Prt\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function DIG\_Prt\_Status% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 Declare Function DIG\_SCAN\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d%, ByVal e%)  
 Declare Function Get\_DAQ\_Device\_Info% Lib "nidaq32.dll" (ByVal a%, ByVal b&, c&)  
 5 Declare Function Get\_DAQ\_Event% Lib "nidaq32.dll" (ByVal a&, b%, c%, d%, e&)  
 Declare Function Get\_NI\_DAQ\_Version% Lib "nidaq32.dll" (a&)  
 Declare Function GPCTR\_Config\_Buffer% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&, e As Any)  
 10 Declare Function GPCTR\_Read\_Buffer% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&, ByVal e&, ByVal f#, g&, h&)  
 Declare Function Line\_Change\_Attribute% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, ByVal d&)  
 Declare Function GPCTR\_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&)  
 Declare Function GPCTR\_Set\_Application% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&)  
 Declare Function GPCTR\_Watch% Lib "nidaq32.dll" (ByVal a%, ByVal b&, ByVal c&, d&)  
 15 Declare Function ICTR\_Read% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 Declare Function ICTR\_Reset% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function ICTR\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)  
 Declare Function Init\_DA\_Brds% Lib "nidaq32.dll" (ByVal a%, b%)  
 20 Declare Function Lab\_ISCAN\_Check% Lib "nidaq32.dll" (ByVal a%, b%, c&, d%)  
 Declare Function Lab\_ISCAN\_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d As Any, ByVal e&, ByVal f#, ByVal g#, h%)  
 Declare Function Lab\_ISCAN\_Start% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d As Any, ByVal e&, ByVal f%, ByVal g%, ByVal h%)  
 25 Declare Function Lab\_ISCAN\_to\_Disk% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d\$, ByVal e&, ByVal f#, ByVal g#, ByVal h%)  
 Declare Function LPM16\_Calibrate% Lib "nidaq32.dll" (ByVal a%)  
 Declare Function MIO\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function Peek\_DAQ\_Event% Lib "nidaq32.dll" (ByVal a&, b%, c%, d%, e&)  
 30 Declare Function REG\_Level\_Read% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c&)  
 Declare Function REG\_Level\_Write% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c&, ByVal d&, e&)  
 Declare Function RTSI\_Clear% Lib "nidaq32.dll" (ByVal a%)  
 Declare Function RTSI\_Clock% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function RTSI\_Conn% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 35 Declare Function RTSI\_DisConn% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function SC\_2040\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function SCXI\_Demux% Lib "nidaq32.dll" (a%, ByVal b&, ByVal c%, ByVal d%)  
 Declare Function SCXI\_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, e As Any, ByVal f&, ByVal g#, ByVal h%)  
 40 Declare Function SCXI\_Sequence\_Demux% Lib "nidaq32.dll" (ByVal a%, b%, ByVal c&, d%, ByVal e%, f%, g&)  
 Declare Function SCXI\_Sequence\_Retrieve% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 Declare Function SCXI\_Sequence\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, e%, f%, g%)  
 Declare Function SCXI\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%)  
 45 Declare Function SCXI\_Start% Lib "nidaq32.dll" (ByVal a%, b As Any, ByVal c&, ByVal d%, ByVal e%, ByVal f%, ByVal g%)  
 Declare Function SCXI\_to\_Disk% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, ByVal e\$, ByVal f&, ByVal g#, ByVal h#, ByVal i%)  
 Declare Function Calibrate\_1200% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f%, ByVal g%, ByVal h%, ByVal i#, ByVal j#)  
 50 Declare Function SCXI\_AO\_Write% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f#, ByVal g%, h%)  
 Declare Function SCXI\_Cal\_Constants% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f%, ByVal g#, ByVal h%, ByVal i%, ByVal j%, ByVal k#, ByVal l#, ByVal m#, ByVal n#, ByVal o#, p#, q#)  
 55 Declare Function SCXI\_Calibrate% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%, ByVal f#, ByVal g#, ByVal h%, ByVal i%)  
 Declare Function SCXI\_Calibrate\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function SCXI\_Change\_Chan% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 60 Declare Function SCXI\_Set\_Excitation% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e!, f!)  
 Declare Function SCXI\_Configure\_Connection% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function SCXI\_Configure\_Filter% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e#, ByVal f%, ByVal g%, h#)  
 65 Declare Function SCXI\_Get\_Chassis\_Info% Lib "nidaq32.dll" (ByVal a%, b%, c%, d%, e%, f%)  
 Declare Function SCXI\_Get\_Module\_Info% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c&, d%, e%)

09703426 103400

Declare Function SCXI\_Get\_State% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, e%)  
 Declare Function SCXI\_Get\_Status% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, d%)  
 Declare Function SCXI\_Load\_Config% Lib "nidaq32.dll" (ByVal a%)  
 Declare Function SCXI\_MuxCtr\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 5 Declare Function SCXI\_Reset% Lib "nidaq32.dll" (ByVal a%, ByVal b%)  
 Declare Function SCXI\_Scale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e#, ByVal f%,  
 ByVal g%, ByVal h%, ByVal i%, j%, k%)  
 Declare Function SCXI\_SCAN\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, e%, ByVal f%, ByVal g%)  
 10 Declare Function SCXI\_Set\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,  
 ByVal f%, g%, h%, i%)  
 Declare Function SCXI\_Set\_Gain% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#)  
 Declare Function SCXI\_Set\_Input\_Mode% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function SCXI\_Set\_State% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%)  
 15 Declare Function SCXI\_Single\_Chan\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function SCXI\_Track\_Hold\_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function SCXI\_Track\_Hold\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%,  
 ByVal e%, ByVal f%, ByVal g%)  
 Declare Function Select\_Signal% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 20 Declare Function Set\_DAQ\_Device\_Info% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function Timeout\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%)  
 Declare Function WFM\_Chan\_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function WFM\_Check% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%, e%)  
 Declare Function WFM\_ClockRate% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,  
 25 ByVal f%)  
 Declare Function WFM\_DB\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, ByVal d%, ByVal e%,  
 ByVal f%)  
 Declare Function WFM\_DB\_HalfReady% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%)  
 Declare Function WFM\_DB\_Transfer% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d As Any, ByVal e%)  
 30 Declare Function WFM\_from\_Disk% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, ByVal d\$, ByVal e%, ByVal f%,  
 ByVal g%, ByVal h%)  
 Declare Function WFM\_Group\_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)  
 Declare Function WFM\_Group\_Setup% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, ByVal d%)  
 Declare Function WFM\_Load% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d As Any, ByVal e%, ByVal f%,  
 35 ByVal g%)  
 Declare Function WFM\_Op% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d As Any, ByVal e%, ByVal f%,  
 ByVal g%)  
 Declare Function WFM\_Rate% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%, d%)  
 Declare Function WFM\_Scale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, e#, f%)  
 40 Declare Function AI\_Read\_Scan% Lib "nidaq32.dll" (ByVal a%, b%)  
 Declare Function AI\_VRead\_Scan% Lib "nidaq32.dll" (ByVal a%, b%)  
 Declare Function SCXI\_ModuleID\_Read% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)  
 Declare Function USE\_E\_Series% Lib "nidaq32.dll" ()  
 Declare Function USE\_E\_Series\_AI% Lib "nidaq32.dll" ()  
 45 Declare Function USE\_E\_Series\_AO% Lib "nidaq32.dll" ()  
 Declare Function USE\_E\_Series\_DIO% Lib "nidaq32.dll" ()  
 Declare Function USE\_E\_Series\_GPCTR% Lib "nidaq32.dll" ()  
 Declare Function USE\_E\_Series\_GPCTR\_Simple% Lib "nidaq32.dll" ()  
 Declare Function USE\_E\_Series\_Misc% Lib "nidaq32.dll" ()  
 50 Declare Function USE\_E\_Series\_WFM% Lib "nidaq32.dll" ()  
 Declare Function AO\_VScale% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c#, d%)  
 Declare Function GPCTR\_Change\_Parameter% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)  
 Declare Function USE\_E\_Series\_DAQ% Lib "nidaq32.dll" ()  
 Declare Function USE\_MIO% Lib "nidaq32.dll" ()  
 55 Declare Function USE\_LPM% Lib "nidaq32.dll" ()  
 Declare Function USE\_LAB% Lib "nidaq32.dll" ()  
 Declare Function USE\_DIO\_96% Lib "nidaq32.dll" ()  
 Declare Function USE\_DIO\_32F% Lib "nidaq32.dll" ()  
 Declare Function USE\_DIO\_24% Lib "nidaq32.dll" ()  
 60 Declare Function USE\_AO\_610% Lib "nidaq32.dll" ()  
 Declare Function USE\_AO\_2DC% Lib "nidaq32.dll" ()  
 Declare Function DIG\_Trigger\_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%, ByVal e%,  
 ByVal f%, ByVal g%, ByVal h%, ByVal i%)  
 Declare Function SCXI\_Set\_Threshold% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e#)  
 65

```

Declare Function WFM_Set_Clock% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#, ByVal e%,
f%)
Declare Function DAQ_Set_Clock% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c#, ByVal d%, e%)
Declare Function Tio_Select_Signal% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d%)
5 Declare Function Tio_Combine_Signals% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function DIG_In_Prt% Lib "nidaq32.dll" (ByVal a%, ByVal b%, c%)
Declare Function DIG_Out_Prt% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%)
Declare Function AI_Get_Overloaded_Channels% Lib "nidaq32.dll" (ByVal a%, b%, c%)
10 Declare Function Calibrate_TIO% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c%, ByVal d#)
Declare Function DIG_Change_Message_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c$, ByVal
d$, ByVal e%, ByVal f%, ByVal g%)
Declare Function DIG_Change_Message_Control% Lib "nidaq32.dll" (ByVal a%, ByVal b%)
Declare Function DIG_Filter_Config% Lib "nidaq32.dll" (ByVal a%, ByVal b%, ByVal c$, ByVal d#)

```

## 15 NIDAQCNS.INC MODULE

```

*****
*
* This file contains definitions for constants required for some
20 * of the NI-DAQ functions.
*
* You should use symbols defined here in your programs; do not
* use the numerical values.
*
25 * See your NI-DAQ Function Reference Manual for details concerning
* use of constants defined here.
*
*****
Global Const ND_ABOVE_HIGH_LEVEL& = 11020
30 Global Const ND_AC& = 11025
Global Const ND_ACK_REQ_EXCHANGE_GR1& = 11030
Global Const ND_ACK_REQ_EXCHANGE_GR2& = 11035
Global Const ND_ACTIVE& = 11037
35 Global Const ND_ADC_RESOLUTION& = 11040
Global Const ND_AI_CALDAC_COUNT& = 11050
Global Const ND_AI_CHANNEL_COUNT& = 11060
Global Const ND_AI_COUPLING& = 11055
Global Const ND_AI_FIFO_INTERRUPTS& = 11600
40 Global Const ND_ANALOG_FILTER& = 11065
Global Const ND_AO48XDC_SET_POWERUP_STATE& = 42100
Global Const ND_AO_CALDAC_COUNT& = 11070
Global Const ND_AO_CHANNEL_COUNT& = 11080
Global Const ND_AO_EXT_REF_CAPABLE& = 11090
45 Global Const ND_AO_UNIPOLAR_CAPABLE& = 11095
Global Const ND_ARM& = 11100
Global Const ND_ARMED& = 11200
Global Const ND_ATC_OUT& = 11250
Global Const ND_ATTENUATION& = 11260
50 Global Const ND_AUTOINCREMENT_COUNT& = 11300
Global Const ND_AUTOMATIC& = 11400
Global Const ND_AVAILABLE_POINTS& = 11500

Global Const ND_BASE_ADDRESS& = 12100
Global Const ND_BELOW_LOW_LEVEL& = 12130
55 Global Const ND_BOARD_CLOCK& = 12170
Global Const ND_BUFFERED_EVENT_CNT& = 12200
Global Const ND_BUFFERED_PERIOD_MSR& = 12300
Global Const ND_BUFFERED_PULSE_WIDTH_MSR& = 12400
Global Const ND_BUFFERED_SEMI_PERIOD_MSR& = 12500
60 Global Const ND_BURST& = 12600
Global Const ND_BURST_INTERVAL& = 12700

Global Const ND_CAL_CONST_AUTO_LOAD& = 13050
Global Const ND_CALIBRATION_ENABLE& = 13055

```



00442-9240460

Global Const ND\_CALIBRATION\_FRAME\_SIZE& = 13060  
 Global Const ND\_CALIBRATION\_FRAME\_PTR& = 13065  
 Global Const ND\_CJ\_TEMP% = &H8000  
 Global Const ND\_CALGND% = &H8001  
 5 Global Const ND\_CLEAN\_UP& = 13100  
 Global Const ND\_CLOCK\_REVERSE\_MODE\_GR1& = 13120  
 Global Const ND\_CLOCK\_REVERSE\_MODE\_GR2& = 13130  
 Global Const ND\_CONFIG\_MEMORY\_SIZE& = 13150  
 Global Const ND\_CONTINUOUS& = 13160  
 10 Global Const ND\_COUNT& = 13200  
  
 Global Const ND\_COUNTER\_0& = 13300  
 Global Const ND\_COUNTER\_1& = 13400  
 Global Const ND\_COUNTER\_2& = 13310  
 15 Global Const ND\_COUNTER\_3& = 13320  
 Global Const ND\_COUNTER\_4& = 13330  
 Global Const ND\_COUNTER\_5& = 13340  
 Global Const ND\_COUNTER\_6& = 13350  
 Global Const ND\_COUNTER\_7& = 13360  
 20  
 Global Const ND\_COUNTER\_1\_SOURCE& = 13430  
 Global Const ND\_COUNT\_AVAILABLE& = 13450  
 Global Const ND\_COUNT\_DOWN& = 13465  
 Global Const ND\_COUNT\_UP& = 13485  
 25 Global Const ND\_COUNT\_1& = 13500  
 Global Const ND\_COUNT\_2& = 13600  
 Global Const ND\_COUNT\_3& = 13700  
 Global Const ND\_COUNT\_4& = 13800  
 Global Const ND\_CURRENT\_OUTPUT& = 40200  
 30  
 Global Const ND\_DAC\_RESOLUTION& = 13950  
 Global Const ND\_DATA\_TRANSFER\_CONDITION& = 13960  
 Global Const ND\_DATA\_XFER\_MODE\_AI& = 14000  
 Global Const ND\_DATA\_XFER\_MODE\_AO\_GR1& = 14100  
 35 Global Const ND\_DATA\_XFER\_MODE\_AO\_GR2& = 14200  
 Global Const ND\_DATA\_XFER\_MODE\_DIO\_GR1& = 14300  
 Global Const ND\_DATA\_XFER\_MODE\_DIO\_GR2& = 14400  
 Global Const ND\_DATA\_XFER\_MODE\_DIO\_GR3& = 14500  
 Global Const ND\_DATA\_XFER\_MODE\_DIO\_GR4& = 14600  
 40 Global Const ND\_DATA\_XFER\_MODE\_DIO\_GR5& = 14700  
 Global Const ND\_DATA\_XFER\_MODE\_DIO\_GR6& = 14800  
 Global Const ND\_DATA\_XFER\_MODE\_DIO\_GR7& = 14900  
 Global Const ND\_DATA\_XFER\_MODE\_DIO\_GR8& = 15000  
  
 45 Global Const ND\_DATA\_XFER\_MODE\_GPCTR0& = 15100  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR1& = 15200  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR2& = 15110  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR3& = 15120  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR4& = 15130  
 50 Global Const ND\_DATA\_XFER\_MODE\_GPCTR5& = 15140  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR6& = 15150  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR7& = 15160  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR8& = 15165  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR9& = 15170  
 55 Global Const ND\_DATA\_XFER\_MODE\_GPCTR10& = 15175  
 Global Const ND\_DATA\_XFER\_MODE\_GPCTR11& = 15180  
  
 Global Const ND\_DC& = 15250  
 Global Const ND\_DDS\_BUFFER\_SIZE& = 15255  
 60 Global Const ND\_DEVICE\_NAME& = 15260  
 Global Const ND\_DEVICE\_POWER& = 15270  
 Global Const ND\_DEVICE\_SERIAL\_NUMBER& = 15280  
 Global Const ND\_DEVICE\_STATE\_DURING\_SUSPEND\_MODE& = 15290  
 Global Const ND\_DEVICE\_TYPE\_CODE& = 15300  
 65 Global Const ND\_DIGITAL\_FILTER& = 15350  
 Global Const ND\_DIGITAL\_RESTART& = 15375

5

10

15

20

25

30

35

40

45

50

55

60

65



Global Const ND\_INT\_REF\_EXTERN& = 21296  
 Global Const ND\_INT\_CAL\_BUS& = 21295  
 Global Const ND\_INT\_MUX\_BUS& = 21305

5 Global Const ND\_INT\_AI\_GND\_AMP\_0& = 21211  
 Global Const ND\_INT\_AI\_GND\_AMP\_1& = 21212  
 Global Const ND\_INT\_AI\_GND\_AMP\_2& = 21213  
 Global Const ND\_INT\_AI\_GND\_AMP\_3& = 21214  
 Global Const ND\_INT\_AO\_CH\_0\_AMP\_0& = 21231  
 10 Global Const ND\_INT\_AO\_CH\_0\_AMP\_1& = 21232  
 Global Const ND\_INT\_AO\_CH\_0\_AMP\_2& = 21233  
 Global Const ND\_INT\_AO\_CH\_0\_AMP\_3& = 21234  
 Global Const ND\_INT\_AO\_CH\_1\_AMP\_0& = 21241  
 Global Const ND\_INT\_AO\_CH\_1\_AMP\_1& = 21242  
 15 Global Const ND\_INT\_AO\_CH\_1\_AMP\_2& = 21243  
 Global Const ND\_INT\_AO\_CH\_1\_AMP\_3& = 21244  
 Global Const ND\_INT\_AO\_CH\_0\_VS\_REF\_AMP\_0& = 21236  
 Global Const ND\_INT\_AO\_CH\_0\_VS\_REF\_AMP\_1& = 21237  
 Global Const ND\_INT\_AO\_CH\_0\_VS\_REF\_AMP\_2& = 21238  
 20 Global Const ND\_INT\_AO\_CH\_0\_VS\_REF\_AMP\_3& = 21239  
 Global Const ND\_INT\_AO\_CH\_1\_VS\_REF\_AMP\_0& = 21251  
 Global Const ND\_INT\_AO\_CH\_1\_VS\_REF\_AMP\_1& = 21252  
 Global Const ND\_INT\_AO\_CH\_1\_VS\_REF\_AMP\_2& = 21253  
 Global Const ND\_INT\_AO\_CH\_1\_VS\_REF\_AMP\_3& = 21254  
 25 Global Const ND\_INT\_AO\_GND\_VS\_AI\_GND\_AMP\_0& = 21266  
 Global Const ND\_INT\_AO\_GND\_VS\_AI\_GND\_AMP\_1& = 21267  
 Global Const ND\_INT\_AO\_GND\_VS\_AI\_GND\_AMP\_2& = 21268  
 Global Const ND\_INT\_AO\_GND\_VS\_AI\_GND\_AMP\_3& = 21269  
 Global Const ND\_INT\_CM\_REF\_AMP\_0& = 21271  
 30 Global Const ND\_INT\_CM\_REF\_AMP\_1& = 21272  
 Global Const ND\_INT\_CM\_REF\_AMP\_2& = 21273  
 Global Const ND\_INT\_CM\_REF\_AMP\_3& = 21274  
 Global Const ND\_INT\_REF\_AMP\_0& = 21291  
 Global Const ND\_INT\_REF\_AMP\_1& = 21292  
 35 Global Const ND\_INT\_REF\_AMP\_2& = 21293  
 Global Const ND\_INT\_REF\_AMP\_3& = 21294  
  
 Global Const ND\_INTERRUPT\_EVERY\_SAMPLE& = 11700  
 Global Const ND\_INTERRUPT\_HALF\_FIFO& = 11800  
 40 Global Const ND\_IO\_CONNECTOR& = 21300  
  
 Global Const ND\_LEVEL\_SENSITIVE& = 24000  
 Global Const ND\_LINK\_COMPLETE\_INTERRUPTS& = 24010  
 Global Const ND\_LOW& = 24050  
 45 Global Const ND\_LOW\_HYSTERESIS& = 24080  
 Global Const ND\_LOW\_TO\_HIGH& = 24100  
 Global Const ND\_LPT\_DEVICE\_MODE& = 24200  
  
 Global Const ND\_MARKER& = 24500  
 50 Global Const ND\_MARKER\_QUANTUM& = 24550  
 Global Const ND\_MAX\_ARB\_SEQUENCE\_LENGTH& = 24600  
 Global Const ND\_MAX\_FUNC\_SEQUENCE\_LENGTH& = 24610  
 Global Const ND\_MAX\_LOOP\_COUNT& = 24620  
 Global Const ND\_MAX\_NUM\_WAVEFORMS& = 24630  
 55 Global Const ND\_MAX\_SAMPLE\_RATE& = 24640  
 Global Const ND\_MAX\_WFM\_SIZE& = 24650  
 Global Const ND\_MEMORY\_TRANSFER\_WIDTH& = 24700  
 Global Const ND\_MIN\_SAMPLE\_RATE& = 24800  
 Global Const ND\_MIN\_WFM\_SIZE& = 24810  
 60  
 Global Const ND\_NEGATIVE& = 26100  
 Global Const ND\_NEW& = 26190  
 Global Const ND\_NI\_DAQ\_SW\_AREA& = 26195  
 Global Const ND\_NO& = 26200  
 65 Global Const ND\_NO\_STRAIN\_GAUGE& = 26225  
 Global Const ND\_NO\_TRACK\_AND\_HOLD& = 26250

Global Const ND\_NONE& = 26300  
 Global Const ND\_NOT\_APPLICABLE& = 26400  
 Global Const ND\_NUMBER\_DIG\_PORTS& = 26500

5 Global Const ND\_OFF& = 27010  
 Global Const ND\_OFFSET& = 27020  
 Global Const ND\_ON& = 27050  
 Global Const ND\_OTHER& = 27060  
 Global Const ND\_OTHER\_GPCTR\_OUTPUT& = 27300  
 10 Global Const ND\_OTHER\_GPCTR\_TC& = 27400  
 Global Const ND\_OUT\_DATA\_FIFO\_SIZE& = 27070  
 Global Const ND\_OUT\_EXTERNAL\_GATE& = 27080  
 Global Const ND\_OUT\_EXTERNAL\_GATE\_POL& = 27082  
 Global Const ND\_OUT\_START\_TRIGGER& = 27100  
 15 Global Const ND\_OUT\_START\_TRIGGER\_POL& = 27102  
 Global Const ND\_OUT\_UPDATE& = 27200  
 Global Const ND\_OUT\_UPDATE\_POL& = 27202  
 Global Const ND\_OUT\_UPDATE\_CLOCK\_TIMEBASE& = 27210  
 Global Const ND\_OUT\_UPDATE\_CLOCK\_TB\_POL& = 27212  
 20 Global Const ND\_OUTPUT\_ENABLE& = 27220  
 Global Const ND\_OUTPUT\_MODE& = 27230  
 Global Const ND\_OUTPUT\_POLARITY& = 27240  
 Global Const ND\_OUTPUT\_STATE& = 27250  
 Global Const ND\_OUTPUT\_TYPE& = 40000  
 25  
 Global Const ND\_DIGITAL\_PATTERN\_GENERATION& = 28030  
 Global Const ND\_PAUSE& = 28040  
 Global Const ND\_PAUSE\_ON\_HIGH& = 28045  
 Global Const ND\_PAUSE\_ON\_LOW& = 28050  
 30  
 Global Const ND\_PFI\_0& = 28100  
 Global Const ND\_PFI\_1& = 28200  
 Global Const ND\_PFI\_2& = 28300  
 Global Const ND\_PFI\_3& = 28400  
 Global Const ND\_PFI\_4& = 28500  
 35  
 Global Const ND\_PFI\_5& = 28600  
 Global Const ND\_PFI\_6& = 28700  
 Global Const ND\_PFI\_7& = 28800  
 Global Const ND\_PFI\_8& = 28900  
 Global Const ND\_PFI\_9& = 29000  
 40  
 Global Const ND\_PFI\_10& = 50280  
 Global Const ND\_PFI\_11& = 50290  
 Global Const ND\_PFI\_12& = 50300  
 Global Const ND\_PFI\_13& = 50310  
 Global Const ND\_PFI\_14& = 50320  
 45  
 Global Const ND\_PFI\_15& = 50330  
 Global Const ND\_PFI\_16& = 50340  
 Global Const ND\_PFI\_17& = 50350  
 Global Const ND\_PFI\_18& = 50360  
 Global Const ND\_PFI\_19& = 50370  
 50  
 Global Const ND\_PFI\_20& = 50380  
 Global Const ND\_PFI\_21& = 50390  
 Global Const ND\_PFI\_22& = 50400  
 Global Const ND\_PFI\_23& = 50410  
 Global Const ND\_PFI\_24& = 50420  
 55  
 Global Const ND\_PFI\_25& = 50430  
 Global Const ND\_PFI\_26& = 50440  
 Global Const ND\_PFI\_27& = 50450  
 Global Const ND\_PFI\_28& = 50460  
 Global Const ND\_PFI\_29& = 50470  
 60  
 Global Const ND\_PFI\_30& = 50480  
 Global Const ND\_PFI\_31& = 50490  
 Global Const ND\_PFI\_32& = 50500  
 Global Const ND\_PFI\_33& = 50510  
 Global Const ND\_PFI\_34& = 50520  
 65  
 Global Const ND\_PFI\_35& = 50530  
 Global Const ND\_PFI\_36& = 50540

Global Const ND\_PFI\_37& = 50550  
 Global Const ND\_PFI\_38& = 50560  
 Global Const ND\_PFI\_39& = 50570

5 Global Const ND\_PLL\_REF\_FREQ& = 29010  
 Global Const ND\_PLL\_REF\_SOURCE& = 29020  
 Global Const ND\_PRE\_ARM& = 29050  
 Global Const ND\_POSITIVE& = 29100  
 Global Const ND\_PREPARE& = 29200  
 10 Global Const ND\_PROGRAM& = 29300  
 Global Const ND\_PULSE& = 29350  
 Global Const ND\_PULSE\_SOURCE& = 29500  
 Global Const ND\_PULSE\_TRAIN\_GNR& = 29600  
 Global Const ND\_PXI\_BACKPLANE\_CLOCK& = 29900  
 15 Global Const ND\_REGLITCH& = 31000  
 Global Const ND\_RESERVED& = 31100  
 Global Const ND\_RESET& = 31200  
 Global Const ND\_RESUME& = 31250  
 20 Global Const ND\_RETRIG\_PULSE\_GNR& = 31300  
 Global Const ND\_REVISION& = 31350  
 Global Const ND\_RTSI\_0& = 31400  
 Global Const ND\_RTSI\_1& = 31500  
 Global Const ND\_RTSI\_2& = 31600  
 25 Global Const ND\_RTSI\_3& = 31700  
 Global Const ND\_RTSI\_4& = 31800  
 Global Const ND\_RTSI\_5& = 31900  
 Global Const ND\_RTSI\_6& = 32000  
 Global Const ND\_RTSI\_CLOCK& = 32100  
 30 Global Const ND\_SCANCLK& = 32400  
 Global Const ND\_SCANCLK\_LINE& = 32420  
 Global Const ND\_SC\_2040\_MODE& = 32500  
 Global Const ND\_SC\_2043\_MODE& = 32600  
 35 Global Const ND\_SELF\_CALIBRATE& = 32700  
 Global Const ND\_SET\_DEFAULT\_LOAD\_AREA& = 32800  
 Global Const ND\_RESTORE\_FACTORY\_CALIBRATION& = 32810  
 Global Const ND\_SET\_POWERUP\_STATE& = 42100  
 Global Const ND\_SIMPLE\_EVENT\_CNT& = 33100  
 40 Global Const ND\_SINGLE& = 33150  
 Global Const ND\_SINGLE\_PERIOD\_MSR& = 33200  
 Global Const ND\_SINGLE\_PULSE\_GNR& = 33300  
 Global Const ND\_SINGLE\_PULSE\_WIDTH\_MSR& = 33400  
 Global Const ND\_SINGLE\_TRIG\_PULSE\_GNR& = 33500  
 45 Global Const ND\_SOURCE& = 33700  
 Global Const ND\_SOURCE\_POLARITY& = 33800  
 Global Const ND\_STABLE\_10\_MHZ& = 33810  
 Global Const ND\_STEPPED& = 33825  
 Global Const ND\_STRAIN\_GAUGE& = 33850  
 50 Global Const ND\_STRAIN\_GAUGE\_EX0& = 33875  
 Global Const ND\_SUB\_REVISION& = 33900  
 Global Const ND\_SYNC\_DUTY\_CYCLE\_HIGH& = 33930  
 Global Const ND\_SYNC\_OUT& = 33970  
 55 Global Const ND\_TC\_REACHED& = 34100  
 Global Const ND\_THE\_AI\_CHANNEL& = 34400  
 Global Const ND\_TOGGLE& = 34700  
 Global Const ND\_TOGGLE\_GATE& = 34800  
 Global Const ND\_TRACK\_AND\_HOLD& = 34850  
 60 Global Const ND\_TRIG\_PULSE\_WIDTH\_MSR& = 34900  
 Global Const ND\_TRIGGER\_SOURCE& = 34930  
 Global Const ND\_TRIGGER\_MODE& = 34970  
 65 Global Const ND\_UI2\_TC& = 35100  
 Global Const ND\_UP\_DOWN& = 35150  
 Global Const ND\_UP\_TO\_1\_DMA\_CHANNEL& = 35200

Global Const ND\_UP\_TO\_2\_DMA\_CHANNELS& = 35300  
 Global Const ND\_USE\_CAL\_CHAN& = 36000  
 Global Const ND\_USE\_AUX\_CHAN& = 36100  
 Global Const ND\_USER\_EEPROM\_AREA& = 37000  
 5 Global Const ND\_USER\_EEPROM\_AREA\_2& = 37010  
 Global Const ND\_USER\_EEPROM\_AREA\_3& = 37020  
 Global Const ND\_USER\_EEPROM\_AREA\_4& = 37030  
 Global Const ND\_USER\_EEPROM\_AREA\_5& = 37040  
  
 10 Global Const ND\_DSA\_RTSI\_CLOCK\_AD& = 44000  
 Global Const ND\_DSA\_RTSI\_CLOCK\_DA& = 44010  
 Global Const ND\_DSA\_OUTPUT\_TRIGGER& = 44020  
 Global Const ND\_DSA\_INPUT\_TRIGGER& = 44030  
 Global Const ND\_DSA\_SHARC\_TRIGGER& = 44040  
 15 Global Const ND\_DSA\_ANALOG\_TRIGGER& = 44050  
 Global Const ND\_DSA\_HOST\_TRIGGER& = 44060  
 Global Const ND\_DSA\_EXTERNAL\_DIGITAL\_TRIGGER& = 44070  
  
 Global Const ND\_VOLTAGE\_OUTPUT& = 40100  
 20 Global Const ND\_VOLTAGE\_REFERENCE& = 38000  
  
 Global Const ND\_VXI\_SC% = &H2000  
 Global Const ND\_PXI\_SC% = &H2010  
 Global Const ND\_VXIMIO\_SET\_ALLOCATE\_MODE& = 43100  
 25 Global Const ND\_VXIMIO\_USE\_ONBOARD\_MEMORY\_AI& = 43500  
 Global Const ND\_VXIMIO\_USE\_ONBOARD\_MEMORY\_AO& = 43600  
 Global Const ND\_VXIMIO\_USE\_ONBOARD\_MEMORY\_GPCTR& = 43700  
 Global Const ND\_VXIMIO\_USE\_PC\_MEMORY\_AI& = 43200  
 Global Const ND\_VXIMIO\_USE\_PC\_MEMORY\_AO& = 43300  
 30 Global Const ND\_VXIMIO\_USE\_PC\_MEMORY\_GPCTR& = 43400  
  
 Global Const ND\_WFM\_QUANTUM& = 45000  
  
 Global Const ND\_YES& = 39100  
 35 Global Const ND\_3V\_LEVEL& = 43450  
  
 Global Const ND\_WRITE\_MARK& = 50000  
 Global Const ND\_READ\_MARK& = 50010  
 Global Const ND\_BUFFER\_START& = 50020  
 40 Global Const ND\_TRIGGER\_POINT& = 50025  
 Global Const ND\_BUFFER\_MODE& = 50030  
 Global Const ND\_DOUBLE& = 50050  
 Global Const ND\_QUADRATURE\_ENCODER\_X1& = 50070  
 Global Const ND\_QUADRATURE\_ENCODER\_X2& = 50080  
 45 Global Const ND\_QUADRATURE\_ENCODER\_X4& = 50090  
 Global Const ND\_TWO\_PULSE\_COUNTING& = 50100  
 Global Const ND\_LINE\_FILTER& = 50110  
 Global Const ND\_SYNCHRONIZATION& = 50120  
 Global Const ND\_5\_MICROSECONDS& = 50130  
 50 Global Const ND\_1\_MICROSECOND& = 50140  
 Global Const ND\_500\_NANOSECONDS& = 50150  
 Global Const ND\_100\_NANOSECONDS& = 50160  
 Global Const ND\_1\_MILLISECOND& = 50170  
 Global Const ND\_10\_MILLISECONDS& = 50180  
 55 Global Const ND\_100\_MILLISECONDS& = 50190  
  
 Global Const ND\_OTHER\_GPCTR\_SOURCE& = 50580  
 Global Const ND\_OTHER\_GPCTR\_GATE& = 50590  
 60 Global Const ND\_AUX\_LINE& = 50600  
 Global Const ND\_AUX\_LINE\_POLARITY& = 50610  
 Global Const ND\_TWO\_SIGNAL\_EDGE\_SEPARATION\_MSR& = 50630  
 Global Const ND\_BUFFERED\_TWO\_SIGNAL\_EDGE\_SEPARATION\_MSR& = 50640  
 Global Const ND\_SWITCH\_CYCLE& = 50650  
 65 Global Const ND\_INTERNAL\_MAX\_TIMEBASE& = 50660  
 Global Const ND\_PRESCALE\_VALUE& = 50670

Global Const ND\_MAX\_PRESCALE& = 50690  
 Global Const ND\_INTERNAL\_LINE\_0& = 50710  
 Global Const ND\_INTERNAL\_LINE\_1& = 50720  
 Global Const ND\_INTERNAL\_LINE\_2& = 50730  
 5 Global Const ND\_INTERNAL\_LINE\_3& = 50740  
 Global Const ND\_INTERNAL\_LINE\_4& = 50750  
 Global Const ND\_INTERNAL\_LINE\_5& = 50760  
 Global Const ND\_INTERNAL\_LINE\_6& = 50770  
 Global Const ND\_INTERNAL\_LINE\_7& = 50780  
 10 Global Const ND\_INTERNAL\_LINE\_8& = 50790  
 Global Const ND\_INTERNAL\_LINE\_9& = 50800  
 Global Const ND\_INTERNAL\_LINE\_10& = 50810  
 Global Const ND\_INTERNAL\_LINE\_11& = 50820  
 Global Const ND\_INTERNAL\_LINE\_12& = 50830  
 15 Global Const ND\_INTERNAL\_LINE\_13& = 50840  
 Global Const ND\_INTERNAL\_LINE\_14& = 50850  
 Global Const ND\_INTERNAL\_LINE\_15& = 50860  
 Global Const ND\_INTERNAL\_LINE\_16& = 50862  
 Global Const ND\_INTERNAL\_LINE\_17& = 50864  
 20 Global Const ND\_INTERNAL\_LINE\_18& = 50866  
 Global Const ND\_INTERNAL\_LINE\_19& = 50868  
 Global Const ND\_INTERNAL\_LINE\_20& = 50870  
 Global Const ND\_INTERNAL\_LINE\_21& = 50872  
 Global Const ND\_INTERNAL\_LINE\_22& = 50874  
 25 Global Const ND\_INTERNAL\_LINE\_23& = 50876  
  
 30  
 Global Const ND\_START\_TRIGGER& = 51150  
 Global Const ND\_START\_TRIGGER\_POLARITY& = 51151  
  
 35  
 Global Const ND\_COUNTING\_SYNCHRONOUS& = 51200  
 Global Const ND\_SYNCHRONOUS& = 51210  
 Global Const ND\_ASYNCHRONOUS& = 51220  
 Global Const ND\_CONFIGURABLE\_FILTER& = 51230  
 40 Global Const ND\_ENCODER\_TYPE& = 51240  
 Global Const ND\_Z\_INDEX\_ACTIVE& = 51250  
 Global Const ND\_Z\_INDEX\_VALUE& = 51260  
 Global Const ND\_SNAPSHOT& = 51270  
 Global Const ND\_POSITION\_MSR& = 51280  
 45 Global Const ND\_BUFFERED\_POSITION\_MSR& = 51290  
 Global Const ND\_SAVED\_COUNT& = 51300  
 Global Const ND\_READ\_MARK\_H\_SNAPSHOT& = 51310  
 Global Const ND\_READ\_MARK\_L\_SNAPSHOT& = 51320  
 Global Const ND\_WRITE\_MARK\_H\_SNAPSHOT& = 51330  
 50 Global Const ND\_WRITE\_MARK\_L\_SNAPSHOT& = 51340  
 Global Const ND\_BACKLOG\_H\_SNAPSHOT& = 51350  
 Global Const ND\_BACKLOG\_L\_SNAPSHOT& = 51360  
 Global Const ND\_ARMED\_SNAPSHOT& = 51370  
 Global Const ND\_EDGE\_GATED\_FSK& = 51371  
 55 Global Const ND\_SIMPLE\_GATED\_EVENT\_CNT& = 51372  
  
 Global Const ND\_VIDEO\_TYPE& = 51380  
 Global Const ND\_PAL\_B& = 51390  
 Global Const ND\_PAL\_G& = 51400  
 60 Global Const ND\_PAL\_H& = 51410  
 Global Const ND\_PAL\_I& = 51420  
 Global Const ND\_PAL\_D& = 51430  
 Global Const ND\_PAL\_N& = 51440  
 Global Const ND\_PAL\_M& = 51450  
 65 Global Const ND\_NTSC\_M& = 51460  
 Global Const ND\_COUNTER\_TYPE& = 51470



```

Global Const ND_NI_TIO& = 51480
Global Const ND_AM9513& = 51490
Global Const ND_STC& = 51500
Global Const ND_8253& = 51510
5 Global Const ND_A_HIGH_B_HIGH& = 51520
Global Const ND_A_HIGH_B_LOW& = 51530
Global Const ND_A_LOW_B_HIGH& = 51540
Global Const ND_A_LOW_B_LOW& = 51550
Global Const ND_Z_INDEX_RELOAD_PHASE& = 51560
10 Global Const ND_UPDOWN_LINE& = 51570
Global Const ND_DEFAULT_PFI_LINE& = 51580
Global Const ND_BUFFER_SIZE& = 51590
Global Const ND_ELEMENT_SIZE& = 51600
Global Const ND_NUMBER_GP_COUNTERS& = 51610
15 Global Const ND_BUFFERED_TIME_STAMPING& = 51620
Global Const ND_TIME_0_DATA_32& = 51630
Global Const ND_TIME_8_DATA_24& = 51640
Global Const ND_TIME_16_DATA_16& = 51650
Global Const ND_TIME_24_DATA_8& = 51660
20 Global Const ND_TIME_32_DATA_32& = 51670
Global Const ND_TIME_48_DATA_16& = 51680
Global Const ND_ABSOLUTE& = 51690
Global Const ND_RELATIVE& = 51700
Global Const ND_TIME_DATA_SIZE& = 51710
25 Global Const ND_TIME_FORMAT& = 51720
Global Const ND_HALT_ON_OVERFLOW& = 51730
Global Const ND_OVERLAY_RTSI_ON_PFI_LINES& = 51740
Global Const ND_STOP_TRIGGER& = 51750
Global Const ND_TS_INPUT_MODE& = 51760
30 Global Const ND_BOTH_EDGES& = 51770

Global Const ND_CLOCK_0& = 51780
Global Const ND_CLOCK_1& = 51790
Global Const ND_CLOCK_2& = 51800
35 Global Const ND_CLOCK_3& = 51810
Global Const ND_SYNCHRONIZATION_LINE& = 51820
Global Const ND_TRANSFER_METHOD& = 51830
Global Const ND_SECONDS& = 51840
Global Const ND_PRECISION& = 51850
40 Global Const ND_NANO_SECONDS& = 51860
Global Const ND_SYNCHRONIZATION_METHOD& = 51870
Global Const ND_PULSE_PER_SECOND& = 51880
Global Const ND_IRIG_B& = 51890
Global Const ND_SIMPLE_TIME_MSR& = 51900
45 Global Const ND_SINGLE_TIME_MSR& = 51910
Global Const ND_BUFFERED_TIME_MSR& = 51920
Global Const ND_DMA& = 51930

```

---

NIDAQERR.INC Module

```

50 *****
* nidaqerr.inc *
* header file for platform-independent ni-daq errors/warnings *
* *
* NOTE: *
55 * You should use symbols defined here in your programs; do not *
* use the numerical values. *
* *
* Warnings are returned as positive numbers. For example *
* overWriteError may be returned as a warning and its value *
60 * would be -(overWriteError). *
* *
* THIS FILE IS AUTOMATICALLY GENERATED FROM A DATABASE: DO NOT EDIT *
* *
*****
65 Global Const noError = 0

```

	Global Const syntaxError = -10001	' An error was detected in the input string; the arrangement or ordering ...
5	Global Const semanticsError = -10002	' An error was detected in the input string; the syntax of the string is ...
	Global Const invalidValueError = -10003	' The value of a numeric parameter is invalid.
	Global Const valueConflictError = -10004	' The value of a numeric parameter is inconsistent with another one, and ...
10	Global Const badDeviceError = -10005	' The device is invalid.
	Global Const badLineError = -10006	' The line is invalid.
	Global Const badChanError = -10007	' A channel, port, or counter is out of range for the device type or device ...
	Global Const badGroupError = -10008	' The group is invalid.
	Global Const badCounterError = -10009	' The counter is invalid.
15	Global Const badCountError = -10010	' The count is too small or too large for the specified counter, or the ...
	Global Const badIntervalError = -10011	' The analog input scan rate is too fast for the number of channels and ...
20	Global Const badRangeError = -10012	' The analog input or analog output voltage or current range is invalid ...
	Global Const badErrorCodeError = -10013	' The driver returned an unrecognized or unlisted error code.
	Global Const groupTooLargeError = -10014	' The group size is too large for the board.
	Global Const badTimeLimitError = -10015	' The time limit is invalid.
25	Global Const badReadCountError = -10016	' The read count is invalid.
	Global Const badReadModeError = -10017	' The read mode is invalid.
	Global Const badReadOffsetError = -10018	' The offset is unreachable.
	Global Const badClkFrequencyError = -10019	' The frequency is invalid.
	Global Const badTimebaseError = -10020	' The timebase is invalid.
30	Global Const badLimitsError = -10021	' The limits are beyond the range of the board.
	Global Const badWriteCountError = -10022	' Your data array contains an incomplete update, or you are trying to write ...
	Global Const badWriteModeError = -10023	' The write mode is out of range or is disallowed.
	Global Const badWriteOffsetError = -10024	' Adding the write offset to the write mark places the write mark outside ...
35	Global Const limitsOutOfRangeError = -10025	' The requested input limits exceed the board's capability or configuration. ...
	Global Const badBufferSpecificationError = -10026	' The requested number of buffers or the buffer size is not allowed. For ...
40	Global Const badDAQEventError = -10027	' For DAQEvents 0 and 1 general value A must be greater than 0 and less ...
	Global Const badFilterCutoffError = -10028	' The cutoff frequency specified is not valid for this device.
	Global Const obsoleteFunctionError = -10029	' The function you are calling is no longer supported in this version of ...
45	Global Const badBaudRateError = -10030	' The specified baud rate for communicating with the serial port is not ...
	Global Const badChassisIDError = -10031	' The specified baud rate for communicating with the serial port is not ...
	Global Const badModuleSlotError = -10032	' The SCXI module slot that was specified is invalid or corresponds to an ...
50	Global Const invalidWinHandleError = -10033	' The window handle passed to the function is invalid.
	Global Const noSuchMessageError = -10034	' No configured message matches the one you tried to delete.
	Global Const irrelevantAttributeError = -10035	' The specified attribute is not relevant.
	Global Const badYearError = -10036	' The specified year is invalid.
55	Global Const badMonthError = -10037	' The specified month is invalid.
	Global Const badDayError = -10038	' The specified day is invalid.
	Global Const stringTooLongError = -10039	' The specified input string is too long. For instance, DAQScope 5102 devices ...
	Global Const badGroupSizeError = -10040	' The group size is invalid.
60	Global Const badTaskIDError = -10041	' The specified task ID is invalid. For instance, you may have connected ...
	Global Const inappropriateControlCodeError = -10042	' The specified control code is inappropriate for the current configuration ...
	Global Const badDivisorError = -10043	' The specified divisor is invalid.
65	Global Const badPolarityError = -10044	' The specified polarity is invalid.
	Global Const badInputModeError = -10045	' The specified input mode is invalid.
	Global Const badExcitationError = -10046	' The excitation value specified is not valid for this device.

09703426 103400

	Global Const badConnectionTypeError = -10047	'The excitation value specified is not valid for this device.
	Global Const badExcitationTypeError = -10048	'The excitation type specified is not valid for this device.
	Global Const badChanListError = -10050	'There is more than one channel name in the channel list that corresponds ...
5	Global Const badTrigSkipCountError = -10079	'The trigger skip count is invalid.
	Global Const badGainError = -10080	'The gain or gain adjust is invalid.
	Global Const badPretrigCountError = -10081	'The pretrigger sample count is invalid.
	Global Const badPosttrigCountError = -10082	'The posttrigger sample count is invalid.
	Global Const badTrigModeError = -10083	'The trigger mode is invalid.
10	Global Const badTrigCountError = -10084	'The trigger count is invalid.
	Global Const badTrigRangeError = -10085	'The trigger range or trigger hysteresis window is invalid.
	Global Const badExtRefError = -10086	'The external reference is invalid.
	Global Const badTrigTypeError = -10087	'The trigger type is invalid.
	Global Const badTrigLevelError = -10088	'The trigger level is invalid.
15	Global Const badTotalCountError = -10089	'The total count is inconsistent with the buffer size and pretrigger scan ...
	Global Const badRPGError = -10090	'The individual range, polarity, and gain settings are valid but the combination ...
	Global Const badIterationsError = -10091	'You have attempted to use an invalid setting for the iterations parameter. ...
20	Global Const lowScanIntervalError = -10092	'Some devices require a time gap between the last sample in a scan and ...
	Global Const fifoModeError = -10093	'FIFO mode waveform generation cannot be used because at least one condition ...
25	Global Const badCalDACconstError = -10094	'The calDAC constant passed to the function is invalid.
	Global Const badCalStimulusError = -10095	'The calibration stimulus passed to the function is invalid.
	Global Const badCalibrationConstantError = -10096	'The specified calibration constant is invalid.
	Global Const badCalOpError = -10097	'The specified calibration operation is invalid.
	Global Const badCalConstAreaError = -10098	'The specified calibration constant area is invalid. For instance, the ...
30	Global Const badPortWidthError = -10100	'The requested digital port width is not a multiple of the hardware port ...
	Global Const gpctrBadApplicationError = -10120	'Invalid application used.
	Global Const gpctrBadCtrNumberError = -10121	'Invalid counterNumber used.
35	Global Const gpctrBadParamValueError = -10122	'Invalid paramValue used.
	Global Const gpctrBadParamIDError = -10123	'Invalid paramID used.
	Global Const gpctrBadEntityIDError = -10124	'Invalid entityID used.
	Global Const gpctrBadActionError = -10125	'Invalid action used.
	Global Const gpctrSourceSelectError = -10126	'Invalid source selected.
40	Global Const badCountDirError = -10127	'The specified counter does not support the specified count direction.
	Global Const badGateOptionError = -10128	'The specified gating option is invalid.
	Global Const badGateModeError = -10129	'The specified gate mode is invalid.
	Global Const badGateSourceError = -10130	'The specified gate source is invalid.
45	Global Const badGateSignalError = -10131	'The specified gate signal is invalid.
	Global Const badSourceEdgeError = -10132	'The specified source edge is invalid.
	Global Const badOutputTypeError = -10133	'The specified output type is invalid.
	Global Const badOutputPolarityError = -10134	'The specified output polarity is invalid.
	Global Const badPulseModeError = -10135	'The specified pulse mode is invalid.
50	Global Const badDutyCycleError = -10136	'The specified duty cycle is invalid.
	Global Const badPulsePeriodError = -10137	'The specified pulse period is invalid.
	Global Const badPulseDelayError = -10138	'The specified pulse delay is invalid.
	Global Const badPulseWidthError = -10139	'The specified pulse width is invalid.
	Global Const badFOUTportError = -10140	'The specified frequency output (FOUT or FREQ_OUT) port is invalid.
55	Global Const badAutoIncrementModeError = -10141	'The specified autoincrement mode is invalid.
	Global Const badNotchFilterError = -10180	'The specified notch filter is invalid.
	Global Const badMeasModeError = -10181	'The specified measurement mode is invalid.
	Global Const EEPROMreadError = -10200	'Unable to read data from EEPROM.
60	Global Const EEPROMwriteError = -10201	'Unable to write data to EEPROM.
	Global Const EEPROMwriteProtectionError = -10202	'You cannot write into this location or area of your EEPROM because it ...
	Global Const EEPROMInvalidLocationError = -10203	'The specified EEPROM location is invalid.
	Global Const EEPROMInvalidPasswordError = -10204	'The password for accessing the EEPROM is incorrect.
65	Global Const noDriverError = -10240	'The driver interface could not locate or open the driver..

	Global Const oldDriverError = -10241	'One of the driver files or the configuration utility is out of date,
	or ...	
	Global Const functionNotFoundError = -10242	'The specified function is not located in the driver.
5	Global Const configFileError = -10243	'The driver could not locate or open the configuration file, or
	the format ...	
	Global Const deviceInitError = -10244	'The driver encountered a hardware-initialization error while
	attempting ...	
	Global Const osInitError = -10245	'The driver encountered an operating-system error while
	attempting to perform ...	
10	Global Const communicationsError = -10246	'The driver encountered an operating-system error while
	attempting to perform ...	
	Global Const cmosConfigError = -10247	'The CMOS configuration-memory for the device is empty or
	invalid, or the ...	
15	Global Const dupAddressError = -10248	'The base addresses for two or more devices are the same;
	consequently, ...	
	Global Const intConfigError = -10249	'The interrupt configuration is incorrect given the capabilities of
	the ...	
	Global Const dupIntError = -10250	'The interrupt levels for two or more devices are the same.
	Global Const dmaConfigError = -10251	'The DMA configuration is incorrect given the capabilities of
20	the computer/DMA ...	
	Global Const dupDMAError = -10252	'The DMA channels for two or more devices are the same.
	Global Const jumperlessBoardError = -10253	'Unable to find one or more jumperless boards you have
	configured using ...	
25	Global Const DAQCardConfError = -10254	'Cannot configure the DAQCard because 1) the correct
	version of the card ...	
	Global Const remoteChassisDriverInitError = -10255	'There was an error in initializing the driver for Remote
	SCXI.	
	Global Const comPortOpenError = -10256	'There was an error in opening the specified COM port.
30	Global Const baseAddressError = -10257	'Bad base address specified in the configuration utility.
	Global Const dmaChannel1Error = -10258	'Bad DMA channel 1 specified in the configuration utility or
	by the operating ...	
	Global Const dmaChannel2Error = -10259	'Bad DMA channel 2 specified in the configuration utility or
	by the operating ...	
35	Global Const dmaChannel3Error = -10260	'Bad DMA channel 3 specified in the configuration utility or
	by the operating ...	
	Global Const userModeToKernelModeCallError = -10261	'The user mode code failed when calling the kernel
	mode code.	
	Global Const noConnectError = -10340	'No RTSI or PFI signal/line is connected, or the specified
40	signal and the ...	
	Global Const badConnectError = -10341	'The RTSI or PFI signal/line cannot be connected as specified.
	Global Const multConnectError = -10342	'The specified RTSI signal is already being driven by a RTSI
	line, or the ...	
	Global Const SCXIConfigError = -10343	'The specified SCXI configuration parameters are invalid, or
45	the function ...	
	Global Const chassisSynchedError = -10344	'The Remote SCXI unit is not synchronized with the host.
	Reset the chassis ...	
	Global Const chassisMemAllocError = -10345	'The required amount of memory cannot be allocated on the
	Remote SCXI unit ...	
50	Global Const badPacketError = -10346	'The packet received by the Remote SCXI unit is invalid.
	Check your serial ...	
	Global Const chassisCommunicationError = -10347	'There was an error in sending a packet to the remote
	chassis. Check your ...	
	Global Const waitingForReprogError = -10348	'The Remote SCXI unit is in reprogramming mode and is
55	waiting for reprogramming ...	
	Global Const SCXIModuleTypeConflictError = -10349	'The module ID read from the SCXI module conflicts
	with the configured ...	
	Global Const CannotDetermineEntryModuleError = -10350	'Neither an SCXI entry module (i.e.: the SCXI
	module cabled to the measurement ...	
60	Global Const DSPInitError = -10360	'The DSP driver was unable to load the kernel for its operating
	system.	
	Global Const badScanListError = -10370	'The scan list is invalid; for example, you are mixing AMUX-
	64T channels ...	
	Global Const invalidSignalSrcError = -10380	'The specified signal source is invalid for the selected signal
	name.	
65	Global Const invalidSignalNameError = -10381	'The specified signal name is invalid.

00103426 103400

	Global Const invalidSrcSpecError = -10382 source or ...	'The specified source specification is invalid for the signal
	Global Const invalidSignalDestError = -10383	'The specified signal destination is invalid.
5	Global Const userOwnedRsrcError = -10400 accessed or ...	'The specified resource is owned by the user and cannot be
	Global Const unknownDeviceError = -10401 the driver ...	'The specified device is not a National Instruments product,
	Global Const deviceNotFoundError = -10402 the driver ...	'The specified device is not a National Instruments product,
10	Global Const deviceSupportError = -10403 (the driver ...	'The specified device does not support the requested action
	Global Const noLineAvailError = -10404	'No line is available.
	Global Const noChanAvailError = -10405	'No channel is available.
	Global Const noGroupAvailError = -10406	'No group is available.
15	Global Const lineBusyError = -10407	'The specified line is in use.
	Global Const chanBusyError = -10408	'The specified channel is in use.
	Global Const groupBusyError = -10409	'The specified group is in use.
	Global Const relatedLCGBusyError = -10410 configures ...	'A related line, channel, or group is in use; if the driver
20	Global Const counterBusyError = -10411	'The specified counter is in use.
	Global Const noGroupAssignError = -10412 cannot be assigned ...	'No group is assigned, or the specified line or channel
	Global Const groupAssignError = -10413 is already ...	'A group is already assigned, or the specified line or channel
25	Global Const reservedPinError = -10414 configured only ...	'The selected signal requires a pin that is reserved and
	Global Const externalMuxSupportError = -10415 external multiplexer ...	'This function does not support your DAQ device when an
30	Global Const sysOwnedRsrcError = -10440 accessed or ...	'The specified resource is owned by the driver and cannot be
	Global Const memConfigError = -10441 mode, or ...	'No memory is configured to support the current data-transfer
	Global Const memDisabledError = -10442 current addressing ...	'The specified memory is disabled or is unavailable given the
35	Global Const memAlignmentError = -10443 data-transfer ...	'The transfer buffer is not aligned properly for the current
	Global Const memFullError = -10444 memory is available ...	'No more system memory is available on the heap, or no more
40	Global Const memLockError = -10445 On PC AT machines, ...	'The transfer buffer cannot be locked into physical memory.
	Global Const memPageError = -10446 may require ...	'The transfer buffer contains a page break; system resources
	Global Const memPageLockError = -10447	'The operating environment is unable to grant a page lock.
45	Global Const stackMemError = -10448	'The operating environment is unable to grant a page lock.
	Global Const cacheMemError = -10449 the current ...	'A cache-related error occurred, or caching is not supported in
	Global Const physicalMemError = -10450 memory is located ...	'A hardware error occurred in physical memory, or no
50	Global Const virtualMemError = -10451 virtual ...	'The driver is unable to make the transfer buffer contiguous in
	Global Const noIntAvailError = -10452	'No interrupt level is available for use.
	Global Const intInUseError = -10453	'The specified interrupt level is already in use by another device.
	Global Const noDMAError = -10454	'No DMA controller is available in the system.
55	Global Const noDMAAvailError = -10455	'No DMA channel is available for use.
	Global Const DMAInUseError = -10456 device.	'The specified DMA channel is already in use by another
	Global Const badDMAGroupError = -10457 because it is too small, ...	'DMA cannot be configured for the specified group
60	Global Const diskFullError = -10458	'The storage disk you specified is full.
	Global Const DLLInterfaceError = -10459 error.	'The NI-DAQ DLL could not be called due to an interface
	Global Const interfaceInteractionError = -10460 compatibility library ...	'You have mixed VIs from the DAQ library and the _DAQ
65	Global Const resourceReservedError = -10461 been reserved ...	'The specified resource is unavailable because it has already

	Global Const resourceNotReservedError = -10462	'The specified resource is unavailable because it has already been reserved ...
	Global Const mdResourceAlreadyReservedError = -10463	'Another entity has already reserved the requested resource.
5	Global Const mdResourceReservedError = -10464	'Another entity has already reserved the requested resource.
	Global Const mdResourceNotReservedError = -10465	'Attempting to lift a reservation off a resource that previously had no ...
10	Global Const mdResourceAccessKeyError = -10466	'The requested operation cannot be performed because the key supplied is ...
	Global Const mdResourceNotRegisteredError = -10467	'The resource requested is not registered with the minidriver.
	Global Const muxMemFullError = -10480	'The resource requested is not registered with the minidriver.
15	Global Const bufferNotInterleavedError = -10481	'You must provide a single buffer of interleaved data, and the channels ...
	Global Const SCXIModuleNotSupportedError = -10540	'You must provide a single buffer of interleaved data, and the channels ...
	Global Const TRIG1ResourceConflict = -10541	'CTRB1 will drive COUTB1, however CTRB1 will also drive TRIG1. This may ...
20	Global Const matrixTerminalBlockError = -10542	'This function requires that no Matrix terminal block is configured with ...
	Global Const noMatrixTerminalBlockError = -10543	'This function requires that some matrix terminal block is configured with ...
25	Global Const invalidMatrixTerminalBlockError = -10544	'The type of matrix terminal block configured will not allow proper operation ...
	Global Const invalidDSPHandleError = -10560	'The DSP handle input is not valid .
	Global Const DSPDataPathBusyError = -10561	'Either DAQ or WFM can use a PC memory buffer, but not both at the same ...
30	Global Const noSetupError = -10600	'No setup operation has been performed for the specified resources. Or, ...
	Global Const multSetupError = -10601	'No setup operation has been performed for the specified resources. Or, ...
	Global Const noWriteError = -10602	'No output data has been written into the transfer buffer.
35	Global Const groupWriteError = -10603	'The output data associated with a group must be for a single channel or ...
	Global Const activeWriteError = -10604	'Once data generation has started, only the transfer buffers originally ...
	Global Const endWriteError = -10605	'No data was written to the transfer buffer because the final data block ...
40	Global Const notArmedError = -10606	'The specified resource is not armed.
	Global Const armedError = -10607	'The specified resource is already armed.
	Global Const noTransferInProgError = -10608	'No transfer is in progress for the specified resource.
	Global Const transferInProgError = -10609	'A transfer is already in progress for the specified resource, or the operation ...
45	Global Const transferPauseError = -10610	'A single output channel in a group may not be paused if the output data ...
	Global Const badDirOnSomeLinesError = -10611	'Some of the lines in the specified channel are not configured for the ...
50	Global Const badLineDirError = -10612	'The specified line does not support the specified transfer direction.
	Global Const badChanDirError = -10613	'The specified channel does not support the specified transfer direction, ...
	Global Const badGroupDirError = -10614	'The specified group does not support the specified transfer direction.
55	Global Const masterClkError = -10615	'The clock configuration for the clock master is invalid.
	Global Const slaveClkError = -10616	'The clock configuration for the clock slave is invalid.
	Global Const noClkSrcError = -10617	'No source signal has been assigned to the clock resource.
	Global Const badClkSrcError = -10618	'The specified source signal cannot be assigned to the clock resource.
60	Global Const multClkSrcError = -10619	'A source signal has already been assigned to the clock resource.
	Global Const noTrigError = -10620	'No trigger signal has been assigned to the trigger resource.
	Global Const badTrigError = -10621	'No trigger signal has been assigned to the trigger resource.
65	Global Const preTrigError = -10622	'The pretrigger mode is not supported or is not available in the current ...
	Global Const postTrigError = -10623	'No posttrigger source has been assigned.

	Global Const delayTrigError = -10624	'The delayed trigger mode is not supported or is not available in the current ...
	Global Const masterTrigError = -10625	'The trigger configuration for the trigger master is invalid.
	Global Const slaveTrigError = -10626	'The trigger configuration for the trigger slave is invalid.
5	Global Const noTrigDrvError = -10627	'No signal has been assigned to the trigger resource.
	Global Const multTrigDrvError = -10628	'A signal has already been assigned to the trigger resource.
	Global Const invalidOpModeError = -10629	'The specified operating mode is invalid, or the resources have not been ...
10	Global Const invalidReadError = -10630	'The parameters specified to read data were invalid in the context of the ...
	Global Const noInfiniteModeError = -10631	'Continuous input or output transfers are not allowed in the current operating ...
	Global Const someInputsIgnoredError = -10632	'Certain inputs were ignored because they are not relevant in the current ...
15	Global Const invalidRegenModeError = -10633	'The specified analog output regeneration mode is not allowed for this ...
	Global Const noContTransferInProgressError = -10634	'No continuous (double buffered) transfer is in progress for the specified ...
20	Global Const invalidSCXIOPModeError = -10635	'Either the SCXI operating mode specified in a configuration call is invalid, ...
	Global Const noContWithSynchError = -10636	'You cannot start a continuous (double-buffered) operation with a synchronous ...
	Global Const bufferAlreadyConfigError = -10637	'Attempted to configure a buffer after the buffer had already been configured. ...
25	Global Const badClkDestError = -10638	'The clock cannot be assigned to the specified destination.
	Global Const rangeBadForMeasModeError = -10670	'The input range is invalid for the configured measurement mode.
	Global Const autozeroModeConflictError = -10671	'Autozero cannot be enabled for the configured measurement mode.
30	Global Const badChanGainError = -10680	'All channels of this board must have the same gain.
	Global Const badChanRangeError = -10681	'All channels of this board must have the same range.
	Global Const badChanPolarityError = -10682	'All channels of this board must be the same polarity.
	Global Const badChanCouplingError = -10683	'All channels of this board must have the same coupling.
35	Global Const badChanInputModeError = -10684	'All channels of this board must have the same input mode.
	Global Const clkExceedsBrdsMaxConvRateError = -10685	'The clock rate exceeds the board's recommended maximum rate.
	Global Const scanListInvalidError = -10686	'A configuration change has invalidated the scan list.
40	Global Const bufferInvalidError = -10687	'A configuration change has invalidated the acquisition buffer, or an acquisition ...
	Global Const noTrigEnabledError = -10688	'The number of total scans and pretrigger scans implies that a triggered ...
	Global Const digitalTrigBError = -10689	'Digital trigger B is illegal for the number of total scans and pretrigger ...
45	Global Const digitalTrigAandBError = -10690	'This board does not allow digital triggers A and B to be enabled at the ...
	Global Const extConvRestrictionError = -10691	'This board does not allow an external sample clock with an external scan ...
50	Global Const chanClockDisabledError = -10692	'This board does not allow an external sample clock with an external scan ...
	Global Const extScanClockError = -10693	'You cannot use an external scan clock when doing a single scan of a single ...
	Global Const unsafeSamplingFreqError = -10694	'The scan rate is above the maximum or below the minimum for the hardware, ...
55	Global Const DMAnotAllowedError = -10695	'You have set up an operation that requires the use of interrupts. DMA ...
	Global Const multiRateModeError = -10696	'Multi-rate scanning cannot be used with the AMUX-64, SCXI, or pretriggered ...
60	Global Const rateNotSupportedError = -10697	'Unable to convert your timebase/interval pair to match the actual hardware ...
	Global Const timebaseConflictError = -10698	'You cannot use this combination of scan and sample clock timebases for ...
	Global Const polarityConflictError = -10699	'You cannot use this combination of scan and sample clock source polarities ...
65	Global Const signalConflictError = -10700	'You cannot use this combination of scan and convert clock signal sources ...

	Global Const noLaterUpdateError = -10701	' The call had no effect because the specified channel had not been set ...
	Global Const prePostTriggerError = -10702	' Pretriggering and posttriggering cannot be used simultaneously on the ...
5	Global Const noHandshakeModeError = -10710	' The specified port has not been configured for handshaking.
	Global Const noEventCtrError = -10720	' The specified counter is not configured for event-counting operation.
10	Global Const SCXITrackHoldError = -10740	' A signal has already been assigned to the SCXI track-and-hold trigger ...
	Global Const sc2040InputModeError = -10780	' When you have an SC2040 attached to your device, all analog input channels ...
	Global Const outputTypeMustBeVoltageError = -10781	' When you have an SC2040 attached to your device, all analog input channels ...
15	Global Const sc2040HoldModeError = -10782	' The specified operation cannot be performed with the SC-2040 configured ...
	Global Const calConstPolarityConflictError = -10783	' Calibration constants in the load area have a different polarity from ...
20	Global Const timeOutError = -10800	' The operation could not complete within the time limit.
	Global Const calibrationError = -10801	' An error occurred during the calibration process. Possible reasons for ...
	Global Const dataNotAvailError = -10802	' The requested amount of data has not yet been acquired.
	Global Const transferStoppedError = -10803	' The on-going transfer has been stopped. This is to prevent regeneration ...
25	Global Const earlyStopError = -10804	' The transfer stopped prior to reaching the end of the transfer buffer.
	Global Const overRunError = -10805	' The clock rate is faster than the hardware can support. An attempt to ...
30	Global Const noTrigFoundError = -10806	' No trigger value was found in the input transfer buffer.
	Global Const earlyTrigError = -10807	' The trigger occurred before sufficient pretrigger data was acquired.
	Global Const LPTcommunicationError = -10808	' The trigger occurred before sufficient pretrigger data was acquired.
35	Global Const gateSignalError = -10809	' Attempted to start a pulse width measurement with the pulse in the phase ...
	Global Const internalDriverError = -10810	' An unexpected error occurred inside the driver when performing this given ...
	Global Const softwareError = -10840	' The contents or the location of the driver file was changed between accesses ...
40	Global Const firmwareError = -10841	' The firmware does not support the specified operation, or the firmware ...
	Global Const hardwareError = -10842	' The hardware is not responding to the specified operation, or the response ...
45	Global Const underFlowError = -10843	' Because of system and/or bus-bandwidth limitations, the driver could not ...
	Global Const underWriteError = -10844	' Your application was unable to deliver data to the background generation ...
	Global Const overFlowError = -10845	' Because of system and/or bus-bandwidth limitations, the driver could not ...
50	Global Const overWriteError = -10846	' Your application was unable to retrieve data from the background acquisition ...
	Global Const dmaChainingError = -10847	' New buffer information was not available at the time of the DMA chaining ...
55	Global Const noDMACountAvailError = -10848	' The driver could not obtain a valid reading from the transfer-count register ...
	Global Const OpenFileError = -10849	' The configuration file or DSP kernel file could not be opened.
	Global Const closeFileError = -10850	' Unable to close a file.
	Global Const fileSeekError = -10851	' Unable to seek within a file.
60	Global Const readFileError = -10852	' Unable to read from a file.
	Global Const writeFileError = -10853	' Unable to write to a file.
	Global Const miscFileError = -10854	' An error occurred accessing a file.
	Global Const osUnsupportedError = -10855	' NI-DAQ does not support the current operation on this particular version ...
65	Global Const osError = -10856	' An unexpected error occurred from the operating system while performing ...



Global Const internalKernelError = -10857 ' An unexpected error occurred inside the kernel of the device while performing ...

Global Const hardwareConfigChangedError = -10858 ' The system has reconfigured the device and has invalidated the existing ...

5 Global Const updateRateChangeError = -10880 ' A change to the update rate is not possible at this time because 1) when ...

Global Const partialTransferCompleteError = -10881 ' You cannot do another transfer after a successful partial transfer.

10 Global Const daqPollDataLossError = -10882 ' The data collected on the Remote SCXI unit was overwritten before it could ...

Global Const wfmPollDataLossError = -10883 ' New data could not be transferred to the waveform buffer of the Remote ...

Global Const pretrigReorderError = -10884 ' Could not rearrange data after a pretrigger acquisition completed.

15 Global Const overLoadError = -10885 ' The input signal exceeded the input range of the ADC.

Global Const gpctrDataLossError = -10920 ' One or more data points may have been lost during buffered GPCTR operations ...

Global Const chassisResponseTimeoutError = -10940 ' No response was received from the Remote SCXI unit within the specified ...

20 Global Const reprogrammingFailedError = -10941 ' Reprogramming the Remote SCXI unit was unsuccessful. Please try again.

Global Const invalidResetSignatureError = -10942 ' Reprogramming the Remote SCXI unit was unsuccessful. Please try again.

25 Global Const chassisLockupError = -10943 ' The interrupt service routine on the remote SCXI unit is taking longer ...

\*\*\*\*\*

30 \* Mapping of old errors and warnings to new

\* Warnings

\* dupIOaddrRange -(dupAddressError)

35 \* dupIntLevels -(dupIntError)

\* dupDMALevels -(dupDMAError)

\* readOutputPort -(badChanDirError)

\* calibrationErr -(calibrationError)

\* noPreTrigUnwrap -(memFullError)

40 \* relatedPortBusy -(relatedLCGBusyError)

\* readOutputLine -(badDirOnSomeLinesError)

\* outOnSomeInLines -(badDirOnSomeLinesError)

\* inOnSomeOutLines -(badDirOnSomeLinesError)

\* simulOpAcrossChips -(invalidOpModeError)

45 \* overWriteBeforeCopy -(overWriteError)

\* pageBreakinWFbuf -(memPageError)

\* wrongNumConfigBytes -(noSetupError)

\* DMAReprogramming -(memPageError)

\* SCXImoduleTypeConflict -(SCXIModuleTypeConflictError)

50 \* notEnoughExtMem -(memFullError)

\* inputModeConflict -(invalidOpModeError)

\* SCXIConfigWarning -(SCXIConfigError)

\* messageIntervalTooLong -(badDAQEventError)

\* logicalDeviceWarning -(badDeviceError)

55 \* calConstPolarityConflict -(calConstPolarityConflictError)

\* irqConflict -(dupIntError)

\* dmaConflict -(dupDMAError)

\* jumperlessBoardWarning -(jumperlessBoardError)

\* gpctrDataLossWarning -(gpctrDataLossError)

60 \* Errors

\* notOurBrdErr unknownDeviceError

65 \* badBrdNumErr badDeviceError

\* badGainErr badGainError

\* badChanErr badChanError

09703466 103100

	* noSupportErr	deviceSupportError
	* badPortErr	badChanError
	* badOutPortErr	badChanDirError
5	* noLatchModeErr	noHandshakeModeError
	* noGroupAssign	noGroupAssignError
	* badInputValErr	invalidValueError
	* timeOutErr	timeOutError
	* outOfRangeErr	badRangeError
10	* daqInProgErr	transferInProgError
	* counterInUseErr	counterBusyError
	* noDAQErr	noTransferInProgError
	* overFlowErr	overflowError
	* overRunErr	overRunError
15	* badCntErr	badCountError
	* brdTypeErr	deviceSupportError
	* noCountOpErr	noEventCtrError
	* ctrReservedErr	sysOwnedRsrcError
	* portAssignToGrp	groupAssignError
20	* noPortAssignErr	noGroupAssignError
	* badGrpDirErr	badGroupDirError
	* noGrpBlockInProg	noTransferInProgError
	* grpBlockInProg	transferInProgError
	* setLatchWGrpCall	invalidValueError
25	* laterIntUpdateNotSet	noLaterUpdateError
	* wfInProgErr	transferInProgError
	* noWfLoadErr	noWriteError
	* noWfInProgErr	noTransferInProgError
	* badPreTrigCntErr	badPretrigCountError
30	* buffNotFullErr	earlyTrigError
	* prePostTrigErr	prePostTriggerError
	* extConvErr	extConvRestrictionError
	* badSigDirErr	badLineDirError
	* noDbDaqErr	noContTransferInProgError
35	* overWriteErr	overWriteError
	* memErr	memFullError
	* noConfigFile	configFileError
	* badGrpSize	badGroupError
	* intLevelInUse	intInUseError
40	* DMAChanInUse	DMAInUseError
	* multSourceInputErr	multConnectError
	* lowScanIntervalErr	lowScanIntervalError
	* noConnectionErr	noConnectError
	* noPGInProg	noTransferInProgError
45	* PGInProg	transferInProgError
	* grpRateErr	counterBusyError
	* extGateErr	invalidOpModeError
	* openFileErr	openFileError
	* writeFileErr	writeFileError
50	* noDbWvfmErr	noTransferInProgError
	* oldDataErr	transferStoppedError
	* dataNotAvailErr	dataNotAvailError
	* DMATransferCntNotAvail	noDMACountAvailError
	* noLabScanErr	noTransferInProgError
55	* dbOpErr	noContWithSynchError
	* DMADisabledErr	noDMAAvailError
	* invalidConfigErr	cmosConfigError
	* brdIsArmedErr	armedError
	* clockSourceErr	multClkSrcError
60	* noSetupErr	noSetupError
	* extConvDrvErr	multClkSrcError
	* triggerSourceErr	badTrigError
	* noArmErr	notArmedError
	* intDisabledErr	noIntAvailError
65	* keyNotFoundErr	configFileError
	* noTrigEnabledErr	preTrigError
	* digPortReserved	sysOwnedRsrcError

09703426 103100  
09703426 103100

	* RTSlineInUseErr	sysOwnedRsrcError
	* dacUpdateRTSlineInUseErr	sysOwnedRsrcError
	* noRTSlineAvailErr	noLineAvailError
5	* preTrigScansErr	badPretrigCountError
	* postTrigScansErr	badPosttrigCountError
	* scanRateErr	badIntervalError
	* invalidGetErr	invalidReadError
	* calInputOutOfRange	badExtRefError
10	* EEPROMAddrErr	EEPROMreadError
	* EEPROMresponseErr	EEPROMreadError
	* EEPROMreadErr	EEPROMreadError
	* EEPROMwriteErr	EEPROMwriteError
	* calResponseErr	calibrationError
	* calConvergeErr	calibrationError
15	* calDACerr	calibrationError
	* externalCalRefErr	badExtRefError
	* internalCalRefErr	hardwareError
	* badOutLineErr	badLineDirError
20	* relatedPortAssignToGrpBusy	relatedLCGBusyError
	* dacUpdateErr	underFlowError
	* muxMemFullErr	muxMemFullError
	* interlvdDataAlignErr	memAlignmentError
	* cannotAlignBufErr	memAlignmentError
25	* cannotLockBufErr	memLockError
	* cannotPageLockErr	memPageLockError
	* invalidChassisIDErr	badChassisIDError
	* invalidModuleSlotErr	badModuleSlotError
	* configFileErr	configFileError
30	* outdatedVDMADerr	oldDriverError
	* ctrRTSNotAvailErr	lineBusyError
	* dacUpdateRTSNotAvailErr	lineBusyError
	* SCXIConfigErr	SCXIConfigError
	* noDbDigErr	noTransferInProgError
35	* DbDigPartialComplete	transferStoppedError
	* SCXITrackHoldErr	SCXITrackHoldError
	* wvfmGrpAssignErr	groupAssignError
	* chanNotAssignedGrpErr	noGroupAssignError
	* grpLoadErr	groupWriteError
40	* loadAfterStartErr	activeWriteError
	* noUpdateRateErr	noClkSrcError
	* chanPauseErr	transferPauseError
	* DSPInitFailure	DSPInitError
	* DSPDataPathInUse	DSPDataPathBusyError
45	* DSPDAQErr	internalKernelError
	* DSPReserved3	badErrorCodeError
	* DSPReserved4	badErrorCodeError
	* DSPReserved5	badErrorCodeError
	* SCXICommErr	communicationsError
50	* invalidOpModeErr	invalidSCXIOpModeError
	* moduleNotSupported	SCXIModuleNotSupportedError
	* DAQboardNotSupported	deviceSupportError
	* noNIDAQLibErr	noDriverError
	* noNIDAQFuncErr	functionNotFoundError
55	* incompatibleVISRDerr	oldDriverError
	* port1InLatchedModeErr	relatedLCGBusyError
	* invalidMemRegionErr	memLockError
	* fifoModeErr	fifoModeError
	* cannotFreeMemErr	memConfigError
60	* memNotLockedErr	memConfigError
	* invalidWinHandleErr	invalidWinHandleError
	* trigEventNotAvailErr	DMANotAllowedError
	* memTypeNotSupportedErr	memConfigError
	* badChanStrErr	syntaxError
65	* parseErr	syntaxError
	* noSuchMessageErr	noSuchMessageError
	* badChanTypeErr	badChanError

```

5  * badTrigValErr          badDAQEventError
   * notOurDSPHandleErr    invalidDSPHandleError
   * NIDAQInternalErr      internalDriverError
   * preTrigReorderErr     pretrigReorderError
   * badCtrErr             badCounterError
   * invalidCtrErr         badCounterError
   * timedMsgInUseErr      counterBusyError
   * invDAQModeTimedMsgErr DMANotAllowedError
10  * lptCommunicationErr   LPTcommunicationError
   * multiRateAMUXErr      multiRateModeError
   * multiRatePreTrigErr   multiRateModeError
   * functionNotLinkedErr  internalDriverError
   * scanIntervalTooLongErr badIntervalError
   * sampleIntervalTooLongErr badIntervalError
15  * updateIntervalTooLongErr badIntervalError
   * gpctrBadApplicationErr gpctrBadApplicationError
   * gpctrBadCounterNumberErr gpctrBadCounterNumberError
   * gpctrBadParamValueErr  gpctrBadParamValueError
   * gpctrBadParamIdErr     gpctrBadParamIdError
20  * gpctrBadEntityIdErr    gpctrBadEntityIdError
   * gpctrBadActionErr      gpctrBadActionError
   * gpctrBadGateSignalErr  gateSignalError
   * gpctrNotArmedErr       noSetupError
   * gpctrNotResetErr      counterBusyError
25  * gpctrNotProgrammedErr  noSetupError
   * gpctrApplicationNotSetErr noSetupError
   * gpctrBufferNotConfiguredErr bufferInvalidError
   * gpctrCantChangeParameterErr counterBusyError
   * lptProtocolNotSupported LPTcommunicationError
30  * rateNotSupportedErr    rateNotSupportedError
   * timebaseConflictErr    timebaseConflictError
   * polarityConflictErr    polarityConflictError
   * signalConflictErr      signalConflictError
   * baseAddressErr         baseAddressError
35  * interruptLevel1Err     badErrorCodeError
   * interruptLevel2Err     badErrorCodeError
   * dmaChannel1Err         dmaChannel1Error
   * dmaChannel2Err         dmaChannel2Error
40  * openSCManagerErr       badErrorCodeError
   * openNIDAQServiceErr    badErrorCodeError
   * startNIDAQServiceErr   badErrorCodeError
   * criticalResourceConflictErr badErrorCodeError
   * jumperlessBoardErr     jumperlessBoardError
45  * reservedPinErr        reservedPinError
   * bufferNotInterleavedErr bufferNotInterleavedError
   * gpctrInUseErr          counterBusyError
   * gpctrDataLossErr       gpctrDataLossError
   * updateRateChangeErr    updateRateChangeError
50  * gpctrBufferConfiguredErr bufferAlreadyConfigError
   * gpctrBufOprnNotInProgErr noTransferInProgError
   * badFilterFreqErr       badFilterCutoffError
   * sc2040HoldModeErr      sc2040HoldModeError
   * sc2040InputModeErr     sc2040InputModeError
55  * noSC2040ConfigErr     noSetupError
   * DAQCardConfigErr       DAQCardConfigError
   * partialTransferCompleteErr partialTransferCompleteError
   * DMABufferAlignmentErr  memAlignmentError
   * outputTypeMustBeVoltageErr outputTypeMustBeVoltageError
60  * osUnsupportedErr       osUnsupportedError
   * osErr                  osError
   * *****

```

#### NIDEX32 MODULE

\*\*\*\*\*

```

* TITLE:   NIDEx32.bas
*          Header for supporting code module for NI-DAQ Examples
*          (32-bit Visual Basic version)
*
5  * DESCR:   This header file is to be used with any NI-DAQ example
*          program.
*          *****/

10  ' NOTE: must also use nidaq32.bas
    '      and nidaqcns.bas

*
* Constants
*
15  * for IType'
    Global Const WFM_DATA_U8 = 0
    Global Const WFM_DATA_I16 = 2
    Global Const WFM_DATA_F64 = 4
20  Global Const WFM_DATA_U32 = 7

* internal constants - change if needed...
    Global Const WFM_PERIODS = 10
    Global Const WFM_MIN_PTS_IN_PERIOD = 2
25  Global Const WFM_U8_MODULO = 256
    Global Const WFM_I16_AMPL = 2047
    Global Const WFM_F64_AMPL = 4.99

30  * error return codes for NIDAQPlotWaveform and NIDAQMakeBuffer
* these error codes are consistent with CVI error codes
    Global Const NIDAQEX_INVALID_BUFFER = -12
    Global Const NIDAQEX_INVALID_NUMPTS = -14
35  Global Const NIDAQEX_INVALID_TYPE = -53

*
* Function prototypes
*
40  Declare Function NIDAQPlotWaveform Lib "nidex32.dll" (pvBuffer As Any, ByVal INumPts&, ByVal IType&)
    As Integer
    Declare Function NIDAQMakeBuffer Lib "nidex32.dll" (pvBuffer As Any, ByVal INumPts&, ByVal IType&) As
    Integer
45  Declare Function NIDAQErrorHandler Lib "nidex32.dll" (ByVal iStatus%, ByVal strFuncName$, ByVal
    iIgnoreWarning%) As Integer
    Declare Function NIDAQDelay Lib "nidex32.dll" (ByVal dSec#) As Integer
    Declare Function NIDAQYield Lib "nidex32.dll" (ByVal iYieldMode%) As Integer
    Declare Function NIDAQMean Lib "nidex32.dll" (pvBuffer As Any, ByVal INumPts&, ByVal IType&, dMean#)
50  As Integer
    Declare Function NIDAQWaitForKey Lib "nidex32.dll" (ByVal dTimeLimit#) As Integer

55

```

## ABSTRACT

5       The objectives of the work reported here were to design and integrate a communications interface and software procedures (i.e., algorithms) for image processing for a Helmet Mounted Display (HMD) image tester. This is a continuation of a previous effort entitled "Preliminary Design of an Image Tester for Helmet Mounted Display."

10       The proposed image quality tester consists of hardware (including camera, lenses, sensors, and fixtures), and software for image capture and analysis. The interface and image processing algorithms are essential components of this system. The interface bridges the gap between hardware devices and software applications, and thus makes information integration possible. The algorithms process, analyze, and characterize the test pattern information generated by a Helmet Mounted Display (HMD).

20       An interface was designed to probe sensor information and coordinate/synchronize image capture and analysis events. A set of three limited switches was utilized to indicate the presence of an HMD, the position of an HMD relative to a wide-angle camera, and the  
Block 19 continued

25       position of an HMD relative to a narrow-angle camera. These switches are connected to a data acquisition card (DAQCard-DIO-24) using designed circuitry. The sensor on/off states are recorded by the card registers. Software routines (i.e., algorithms) were designed and developed to probe the register status, and then use this information to coordinate/synchronize image characterization events. In order to enhance the flexibility and reduce the complexity of the existing image capture application, a new image capture module was designed.

30       In designing the algorithms, issues such as data collection steps, design specifications, and noise generation were taken into consideration. Three HMD units were utilized to capture image data. Images with noise—such as displacement and variations in orientation and focus—were captured. Statistical approaches such as correlation coefficients and regression analysis were utilized to probe the relationships between performance/image related variables (such as focus) and image gray level variation. Knowledge of such relationships enables the use of image variables to verify and/or predict control variables such as focus resolution. Image measurement specifications  
35       were developed based on analysis of the collected image data. Algorithms for detecting four vertical lines, center point, focus, and boundary are proposed. Examples are given to illustrate how the algorithms work and screenshots of images before and after image processing are shown.

**EXAMPLE IMPLEMENTATION B**

(The following is a substantial duplicate of Hsieh, et al., "Preliminary Design of an Image Quality Tester For Helmet-Mounted Displays," USAARL Report No. 2000-08 (November 1999), the content of which is hereby incorporated by reference in its entirety.

**USAARL Report No. 2000-08**

**Preliminary Design of an Image Quality Tester  
for  
Helmet-Mounted Displays**

**by**

**Sheng-Jen Hsieh**

**Texas A&M University**

**and**

**Clarence E. Rash**

**Aircrew Health and Performance Division**

**and**

**Thomas H. Harding**

**Howard H. Beasley**

**John S. Martin**

**UES, Inc.**

**December 1999**

**Approved for public release, distribution unlimited.**

**U.S. Army Aeromedical Research Laboratory  
Fort Rucker, Alabama 36362-0577**

## Notice

### Qualified requesters

- 5 Qualified requesters may obtain copies from the Defense Technical Information Center (DTIC), Cameron Station, Alexandria, Virginia 22314. Orders will be expedited if placed through the librarian or other person designated to request documents from DTIC.

10 Change of address

Organizations receiving reports from the U.S. Army Aeromedical Research Laboratory on automatic mailing lists should confirm correct address when corresponding about laboratory reports.

15

### Disposition

Destroy this document when it is no longer needed. Do not return it to the originator.

20 Disclaimer

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documents. Citation of trade names in this report does not constitute an official Department of the Army endorsement or approval of the use of such commercial items.

25



Table of contentsPage

5	Introduction	91
	Functionality and operating process	91
	Methodology	92
10	Image capture hardware specification	92
	Test pattern features investigation	95
15	Software prototype design	96
	Image capture module	97
	Image analysis and interpretation module	98
20	Algorithm design	98
	Testing and validation	100
25	Hardware package design	100
	Conclusions and future directions	101
	References	102
30	Appendix A. List of manufacturers	103
	Appendix B. Software prototype program	104

## List of figures

40	B1. The IHU of the AH-64 IHADSS	91
	B2. The IHADSS HDU	91
	B3. Display size	92
45	B4. Test pattern from the IHADSS HMD	92

Table of contents (continued)  
List of figures (continued)

	<u>Page</u>
5	
B5. Flow chart for the HMD prototype tester operation	92
B6. Experimental setup for camera sensitivity analysis	93
10 B7. Sampling locations on the test pattern	93
B8. Plot of photometer and CCD camera data	94
B9. Setup for test pattern measurement	95
15 B10. Test pattern design based on measurement results	95
B11. Replicated test pattern image	95
20 B12. Measurement of luminance of the center lines	96
B13. Center lines measurement with varied focus	96
B14. Designed test pattern with focus on the center lines	96
25 B15. Opening screen of prototype software	97
B16. Image capture module	97
30 B17. Image capture component	97
B18. Image processing component	97
B19. Image analysis and interpretation module	97
35 B20. Tilted test pattern binary images from image analysis module	100
B21. Overall testing results of an HMD	100
40 B22. Tilted test pattern before (left) and after (right) Sober edge detection	100
B23. Investigation of CCD image capture arrangement	101
B24. CAD concept of prototype hardware design	101
45	
List of tables	
Measured data and correlation coefficient from photometer and CCD camera	94

## Introduction

Helmet-mounted displays (HMDs) are a gateway to the pilot for viewing pilotage and fire control imagery. In Army aviation, the AH-64 Apache helicopter uses an HMD system known as the Integrated Helmet and Display Sighting System (IHADSS). The IHADSS consists of various electronic components and a helmet/display system called the Integrated Helmet Unit (IHU). The IHU (Figure B1) includes a helmet, visor housings with visors, miniature cathode ray tube (CRT), and helmet display unit (HDU). The HDU serves as an optical relay device which conveys the image formed on the CRT through a series of lenses, off a beamsplitter (called a combiner), and into the aviator's right eye (Figure B2). The CRT is 1 inch in diameter and uses a P-43 phosphor. The combiner is a multilayer dichroic filter which is maximized for reflectance at the peak emission of the P-43 phosphor.

The U.S. Army is currently developing the next generation reconnaissance aircraft, the RAH-66 Comanche. This aircraft will incorporate an HMD which will be binocular in design. While its final design is still in review, it will basically consist of two image sources (either miniature CRTs or liquid crystal displays) with two sets of optics, delivering imagery to both eyes.

See

**Figure B1. The IHU of the AH-64 IHADSS.**

Currently, there is no existing image quality tester for HMD validation in the field. To maintain system integrity and readiness, and to provide pilots with validated pilotage, navigation, and fire control imagery, there is a need to design and construct an image quality testing tool for the HMD. The objective of this study is to propose and test a design concept for an image quality tester for HMD subsystems. The tester can be used as a validation tool to verify settings for regular flight missions and for preventive maintenance tasks. The first prototype tester will be designed for the AH-64's IHADSS HMD.

**See**

Figure B2. The IHADSS HDU.

### Functionality and operating process

The proposed tester will allow pilots and maintenance personnel to validate the image quality of an HMD. Basic required characteristics include (1) simple design, (2) ease of use, (3) robustness, and (4) accuracy for operations and maintenance. The prototype should be small enough to fit into a brief case, which would include a lap-top, image capture system, and power supply pack.

The IHADSS HMD has a monocular 30-degree vertical by 40-degree horizontal field-of-view (FOV). Future HMDs most likely will have larger FOVs and be binocular in design. HMD corner obscurations are generally permissible and symmetrical for the IHADSS, as illustrated in Figure B3. Since hardware changes to

5 Each strip contains 8 to 10 shades of gray, depending on the contrast ratio. Adjacent  
shades have a square root of 2 differential of brightness. Figure B4 is a snapshot of  
the test pattern captured from the IHADSS HMD. For more detailed discussion of  
the HMD test pattern features, see the Honeywell, Inc. study guide (1985) and  
Harding et al. (1995). For testing this test pattern, the inspection features used by the  
10 image quality tester prototype will include (1) four vertical center lines, (2) one  
horizontal center line, (3) two gray shade patterns (with 8 to 10 shades), and (4) a  
boundary box.

15

20

30

## Methodology

35

## 40

45

10

20

25

Table.

Luminance vs gray level (7 shades): Correlation = 0.983886; Fisher's z = 2.406549;  
Probability = 00006

**See**

Figure B8. Plot of photometer and CCD camera data.

10 In an attempt to capture the test pattern image on the IHADSS fully, several  
different cameras (with standard lenses) were evaluated. However, although the full  
test pattern could be captured, the details of the four vertical center lines could not be  
differentiated. Therefore, a decision was made to use a narrow angle lens to zoom in  
on the center area of the test pattern in order to capture the details of the center lines.  
15 HMDs are also used at night; therefore, the prototype tester--specifically the camera--  
should provide good sensitivity at low luminance levels. First order specifications for  
the required camera were summarized as follows:

1. Sensitivity: #0.005 lux
2. Focus: To infinity
3. Resolution: > 768 x 498 pixels
4. Focal length: ~½ inch
5. Iris: Manual
6. Fields of view: >40 (H) x 30 (V) degrees and ~5 x 3 degrees



observations and illustrates the concept. Findings from the above experiments, such as measurements, luminance ratios, and the content of each feature within the test pattern, were used to create a test pattern image using graphics software. Figure B14 shows an image of such a test pattern using a 5 X 4 degree lens to focus on the center lines of the test pattern. In addition, the ratio of the square root of 2 luminance difference was used to design gray shades ranging from 0 to 255 gray levels.

See

Figure B12. Measurement of luminance of the center lines.

See

Figure B13. Center lines measurement with varied focus.

To emulate potential human errors in setting up the HMD, a set of parameters (including brightness, orientation, spatial adjustment, and contrast) were manipulated and the resulting images captured. These images were used as a basis for creating new image files. These designed images were used to test the software prototype. The experiments were carried out using similar methods. For example, to measure the potential displacement of the test pattern, a camera was mounted facing the HMD. The test pattern was projected onto a video monitor by means of a personal computer (PC). Measurements were taken before and after the spatial adjustments. The maximum adjustments in the upward, downward, left and right directions were 3.57, 2.98, 4.90 and 4.90 degrees, respectively, based on an FOV of 40 x 31 degrees (Harding et al., 1995).

See

Figure B14. Designed test pattern with focus on the center lines.

#### Software prototype design

The software prototype was designed to capture, analyze, and interpret the image against test pattern features such as the four center lines and number of gray shades. Accordingly, the prototype design will require three modules--image acquisition, image analysis and interpretation--as well as on-line user help. Figure B15 shows the modules involved in the prototype. Visual Basic (VB) was used to develop the prototype because of its flexibility in linking and embedding with other commercial software and because it was a powerful toolbox for rapidly prototyping a complicated window. In the following sections, we describe the functionality of each module and how the modules are integrated. Algorithms developed to interpret the image follow. Finally, testing and validation of the code is addressed. The source code for the program can be found in Appendix B.



### Image capture module

The VB Object Linking and Embedding (OLE) capability allows integration of other programs. In this case, the image capture graphics program served as an object which was linked into the VB main program. The graphics program was launched by activating the linked object. Once the object had been activated, the VB main program allowed the user to modify, save, or open documents created by the graphic program in VB's integrated design environment (IDE). After the user was done with the image capture graphics program, control was released to the VB environment. The graphics program itself contained three components: the driver used to activate the image capture card and digitize the video signal into a graphics image format (e.g., bitmap or jpeg); an image processing shell which allowed image manipulation (e.g., sharpening and lightening); and an on-line user manual. Figure B16 shows the opening screen for the image capture module. Figures B17 and B18 show image capture and processing subcomponents.

See

Figure B15. Opening screen of prototype software.

See

Figure B16. Image capture module.

See

Figure B17. Image capture component.

See

Figure B18. Image processing component.

### Image analysis and interpretation module

The image analysis and interpretation module (1) detects the presence of key features such as center lines within the test pattern, (2) compares selected features against the feature specification, and (3) generates findings. VB components were created to provide these functions and to interface with other modules. A subwindow titled "evaluation criteria" was created to analyze and interpret the captured image from an HMD. A few created algorithms were coded in VB to perform the analysis. Other subwindows, such as a directory box and file list boxes were created to allow retrieval of image files for analysis. Finally, an additional subwindow was designed to display the image currently being analyzed. This module also allows access to other modules via a button control. Figure B19 shows the image analysis and interpretation module.

See

Figure B19. Image analysis and interpretation module.

Algorithms were developed to detect various features within the test pattern as described earlier. These are described below:

A. Identify the number of center lines.

**Step 2. Draw multiple lines across X and/or Y axes.**

**Step 4. Store the intersection points in an array with multiple dimensions.**

**Step 6. Develop regression lines to compare the parallel property.**

15

**Note 2: Use of linear regression analysis would make the linear mode robust and insensitive to noise presence.**

**Step 1. Construct a regression line based on all the intercepted points.**

**Step 3. Examine neighboring pixels to see if a W/W/W mask exists.**

**Step 5. Check the distance of neighboring pixels from the regression line using a 3 x 3 area.**

30

**Step 7. Go to step 3.**

**Step 1. Compute a theoretical center as point A.**

**Step 3. Compute the distance between point A and B as  $d$ .**

**Step 5. Construct lines between a given point with points A and B.**

40

**Step 1. Use the center point as a starting point.**

45

**Step 4. Store it in one dimension of the array.**

**Step 5.** If the boundary is not reached, move up or down a given distance, and go to Step 3; else

next.

Step 6. Use of square root of two differences to determine the number of gray shades.

E. Identify boundary lines.

Step 1. Use the center point and boundary ratio to determine the region of the image boundary.

Step 2. Locate a starting point white pixel to use for back tracking the rest of the white pixels for each line segment.

F. Identify the focus setting.

Step 1. Use line scan technique to record the pixels along the center lines.

Step 2. Use the B/W/B mask to identify the separation of lines.

Step 3. Compute the ratio of bottom to mid-peak and peak to valley for all four lines.

Step 4. If the ratio is approximately one, we may conclude that the focus setting is good; or else check the focus setting

Other methods for center point detection exist. However, these were deemed less appropriate for this application. For instance:

Alternate approach #1:

b b b b

Step 1. Use of the mask of bwwwwwb

b b b b

Note: If the orientation of the image is unknown, this method can be time consuming.

Alternate approach #2:

Step 1. Find the center point of each line.

Step 2. Use the averaging method to find the center of all the centers.

Note: This method involves more steps than the proposed one, because you must first find the center of each line and there are four lines to be examined.

Alternate approach #3:

Step 1. Identify the boundary of the image.

Step 2. Use the center of gravity method to find the center of the image.

Figure B20(a-d) shows screens from the image analysis module. Figure B20a shows a binary image of the test pattern after the binary image technique had been applied to the test pattern captured from the HMD. Figure B20b shows the four center lines that were identified from the binary image (Figure B20a). After the center lines had been identified, the image analysis module identified the center point of the image. Figure B20c shows the coordinates (y only shown) of the center point. The image analysis module then determined if the image was tilted or not. Figure B20d displays the tilt angle of the image. The analysis results are summarized and displayed in Figure B21. A primary feature of the image analysis module is to

identify features present in the captured test pattern. The "Sober operator," a well known edge detection technique, is used to identify the boundaries of the features and, thereby, allow the analysis module to determine whether or not the required features are present in the captured test pattern image. Figure B22 shows the same image before and after the Sober operator is applied.

### Testing and validation

To verify the accuracy of the program, language debugging tools, and split-half and back tracking strategies were imposed throughout the coding process. The program results were compared with the simulation results. For example, to check the accuracy of the constructed regression line, the same data points also were analyzed and compared with the results obtained from a statistics package and hand calculation.

See

Figure B20. Tilted test pattern binary images from image analysis module.

See

Figure B21. Overall testing results of an HMD.

See

Figure B22. Tiled test pattern before (left) and after (right) Sober edge detection.

### Hardware package design

A preliminary concept for the hardware package design consists of a display/output module, power supply module, and image capture module. The display/output module should be designed to display/generate inspection results of an HMD test pattern. The power supply module should be designed to provide the voltages needed for the cameras and computer. The design also should include a rechargeable battery pack which will allow the unit to operate in areas without an external power supply. The power supply would be required to provide 12- and 9-volt outputs for the cameras and computer, respectively. Finally, the image capture module must be designed to hold an HMD and two cameras in fixed and contained positions, thereby preventing potential noise that may affect the inspection accuracy. A proposed design is as follows: Two cameras arranged vertically and facing the HMD. [Figure B23 shows one method investigated for aligning the CCD image capture cameras and the HMD.] An inverted HMD fixture will be the most likely one be used in the final concept. The fixture would be mounted with spring return locks on the sides and bottom. The spring return locks will lock the HMD in a fixed position. These locks would prevent the inspection process from continuing if the HMD is not positioned correctly. Once the HMD is in the correct position, a proximity sensor will be used to trigger the image system to start the image capture and interpretation processes. The cover of the image capture module is in the shape of an inverted HMD. It is designed to cover the HMD tightly once it is in the correct

position, and to eliminate any optical noise from the surrounding environment. To enhance the speed of image analysis, an Electronic Programmable Read Only Memory (EPROM) chip, custom programmed to load the executable program for image analysis, could be used. Figure B24 illustrates a preliminary computer aided design (CAD) concept of the hardware prototype design.

### Conclusions and future directions

In this project, a design framework for an image quality tester was proposed and evaluated. Functionality and requirements of the tester were identified. Experiments were conducted to test camera sensitivity and to probe aspects of an HMD test pattern using programmable micro-positioning systems and a CCD camera. Test pattern specifications were developed based on these observations. A strategy for image analysis and interpretation was formed, and algorithms were designed to verify the test pattern of a given HMD against the specifications. A prototype software package was written to inspect the test pattern and verify the effectiveness of the algorithms. Finally, a design framework for a concept hardware package was proposed.

See  
Figure B23. Investigation of CCD image capture arrangement.

See  
Figure B24. CAD concept of prototype hardware design.

To build a brassboard version of a tester, future work must include: (1) fabrication of the hardware design using inverse casting techniques, (2) integration of software and hardware components for a prototype design, (3) field testing of the prototype, (4) incorporation of learning algorithms to increase inspection accuracy, and (5) expansion of functionality from validation to on-line real time interactive adjusting and self-tuning based on a given environmental scenario. From the maintenance perspective, the work can be expanded to self-diagnosis and preventative maintenance (such as life-time prediction).

# References

Avionics Systems Group, Military Avionics Division. 1985. Integrated Helmet and Display Sighting System - Study Guide. St. Louis Park, MN: Honeywell, Inc.

5

Harding, T.H., Beasley, H.H., Martin, J.S. and Rash, C.E. 1995. Physical Evaluation of the Integrated Helmet and Display Sighting System Helmet Display Unit. Fort Rucker, AL: U.S. Army Aero0medical Research Laboratory. USAARL Report No. 95-32.

10

001001 " 924E0460

Appendix A

List of manufacturers.

- 5 Photo Research  
3000 North Hollywood Way  
Burbank, CA 91505

09/03/95 10:00

Appendix B

Software prototype program.

5

001001 " 92420/60



**ABSTRACT**

5       Helmet-mounted displays (HMD's) provide essential pilotage and fire control  
imagery information for pilots. However, image quality testers for HMD field  
performance validation do not currently exist. This research employed techniques  
from imaging analysis and interpretation, and computer-aided design/computer-aided  
manufacturing (CAD/CAM) to investigate a preliminary design for a portable HMD  
image quality tester.

10       For this study, a charge coupled device (CCD) camera and lens were selected.  
Hardware characteristics such as viewing angles in horizontal and vertical positions,  
dynamic working range at day and night, pixel resolution, focal length, and aperture  
ratio were evaluated with regard to HMD functionality. Experiments to evaluate  
camera sensitivity and test pattern merits were conducted using a programmable  
15       micro-positioning system, CCD camera, optical fixtures and benches. Next, the  
relative ratio among features within the image profile was established and an ideal  
image profile and evaluation criteria were established based on the experimental  
results. Third, image processing algorithms and techniques, such as edge detection,  
were developed and applied in test pattern feature detection. A software prototype  
20       including modules for image capture, image analysis and interpretation, and user  
manuals was developed. Finally, a concept hardware package design is proposed.  
This design incorporates a notebook computer with flat panel display to interface with  
the camera and software prototype; and fixtures for the HMD, camera, computer, and  
power supply. This design will allow the tester to be used in the field.

25

004604-9460/60

The foregoing described embodiments depict different components contained within, or connected with, different other components. It is to be understood that such depicted architectures are merely exemplary, and that in fact many other architectures can be implemented which achieve the same functionality. In a conceptual sense, any arrangement of components to achieve the same functionality is effectively “associated” such that the desired functionality is achieved. Hence, any two components herein combined to achieve a particular functionality can be seen as “associated with” each other such that the desired functionality is achieved, irrespective of architectures or intermedial components. Likewise, any two components so associated can also be viewed as being “operably connected”, or “operably coupled”, to each other to achieve the desired functionality.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects and, therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be understood by those within the art that if a specific number of an introduced claim element is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation no such intent is present. For example, as an aid to understanding, the following appended claims may contain usage of the introductory phrases “at least one” and “one or more” to introduce claim elements. However, the use of such phrases should not be construed to imply that the introduction of a claim element by the indefinite articles “a” or “an” limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases “one or more” or “at least one” and indefinite articles such as “a” or “an”; the same holds true for the use of definite articles used to introduce claim elements. In addition, even if a specific number of an introduced claim element is explicitly recited, those skilled in the art will recognize that such recitation should typically be interpreted to mean *at least* the recited number (e.g., the bare recitation of “two elements,” without other modifiers, typically means *at least* two elements, or *two or more* elements).